Reinforcement Learning in Multi-agent Games

A value iteration perspective

Michael Kaisers

March 15, 2008

Abstract

This article investigates the performance of independent reinforcement learners in multi-agent games. Convergence to Nash equilibria and parameter settings for desired learning behavior are discussed for Q-learning, Frequency Maximum Q value (FMQ) learning and lenient Q-learning.
FMQ and lenient Q-learning are shown to outperform regular Q-learning significantly in the context of coordination games with mis-coordination penalties. Furthermore, Q-learning with an ϵ-greedy and FMQ learning with a Boltzmann action selection are shown to scale well to games with one thousand agents.

Keywords: iterated games, reinforcement learning, Q-learning, FMQ-learning, lenient Q-learning

1 Introduction

In this paper value based reinforcement learning algorithms, namely Q-learning and two adaptations, are compared in multi-agent games. Learning in multi-agent environments is significantly more complex than single-agent learning as the dynamics to learn change by the learning process of other agents. This makes predicting learning behavior of learning algorithms in multi-agent environments difficult. They are not only situated in a non-stationary environment but also need to deal with incomplete information and communication limits. In non-stationary environments the Markov property does not hold which makes all proofs of convergence to optimal policies from single-agent learning that are based on that assumption inapplicable. This reduces the theoretical framework available for multi-agent learning. More recently, Evolutionary Game Theory (EGT) with less strong assumptions than classical Game Theory (GT) could be linked to Reinforcement Learning (RL) and provides useful insights into the learning dynamics [14, 4].

The approaches to multi-agent learning vary from joint action space to independent learners. Joint action space learners belong to the class of model based learning while independent learners are model free. Model based learning tries to make use of knowledge about the underlying structure of the problem, but precisely this information is often supposed to be unavailable in games. Due to incomplete information it is attractive to choose for independent learners in multi-agent environments.

Independent multi-agent RL belongs to the class of model free learning - the information is entirely extracted from a numerical reward feedback from the environment. This feedback depends on the action sequence that the agent executes in the environment. As such it can be used to relate rewards to actions and learn a policy that maximizes the reward signal. It has been shown that these learners can be used for action coordination in co-operative and competitive settings [11].

The presented findings survey algorithm performances in different multi-agent games and indicate parameter settings for which desired learning behavior can be achieved. Besides two-agent two-action games, penalty games and coordination games are studied and significant differences in learning performances are pointed out. Concepts like Nash equilibria, Pareto optimality and social welfare are considered, particular attention is devoted to variations of initial conditions, convergence speed and convergence to global or local optima. The learning behaviors of the value iterators Q-learning, FMQ and lenient Q-learning are studied by simulation and analysis supported by visualizations of the learning dynamics. A comparison with policy iterators and learning automata in particular can be performed consulting [6] in which the same analytical means are deployed.

This article is structured as follows: Section 2 provides definitions of games and analytical concepts from GT. Reinforcement learning and the learning algorithms in particular are introduced in Section 3. The tools which are used to investigate learning behaviors in the experiments are shown in Section 4 and obtained results are laid out in Section 5. Section 6 discusses the findings and puts them into context. Finally, Section 7 draws conclusions based on the discussion.

2 Concepts from Game Theory

This section presents the required concepts from GT. It starts with an intuitive and a formal definition of games, gives the means to characterize them and argues for the selected examples that are employed in the experiments. The definitions are based on [3, 4].

Game theory introduced games as formal models to study interactive situations. It emerged from the investigation of strategic conflicts, e.g. in economics and war, and was founded by John von Neumann and Oskar Morgenstern who published the first important book [15] in this discipline together in 1944[1]. Since then, the theory has been enriched by many contributors. Among them John Nash, who introduced what is now known as the Nash Equilibrium (NE) in 1951 and John Maynard Smith, who contributed the notion of Evolutionary Stable Strategies (ESS) in 1973.

2.1 Games

Games model the strategic interaction of players, in the context of reinforcement learning also called agents. Each player has a set of available actions, called pure strategies, and is affected by all other players' actions. More specifically, the player has a preference about the action profile which is the set of actions played by all agents. The formal definition of a game specifies the players, the actions available to them, preferences for outcomes and the information that is available about other players' actions and preferences.

Games can be played once or repeatedly. Iterated play allows the strategy to be dependent on previous outcomes of the game. In other words it allows to learn a strategy from experience.

Traditionally game theory assumes rationality. This means each player is assumed to be absolutely self interested, capable and willing to consider all possible outcomes of the game and selecting the action that maximizes the expected payoff for that player. One of the main criticisms against game theory is the surrealism of that assumption because rationality does not always apply, especially not to humans. However, the learners under investigation are rational hence game theory provides a good framework for the analysis.

There are two representations for games, the extensive and the strategic form. The extensive form describes how the game is played over time in a game tree. The outcome is captured in a single value for each player, the utility, reward or payoff, that denotes the preference of that player for that outcome.

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

Figure 1: The payoff matrix for the Prisoners' dilemma. Player one chooses a row, player two chooses a column, each player can *Deny* or *Confess*. The first number of the selected action combination represents the payoff to player one and the second number the payoff to player two.

Not every game requires this description. Normal form games or strategic games are games of simultaneous action selection. The utility function for each player can be summarized in a matrix that lists his choices against all combinations of opponents' choices. A common notation for two-player normal form games is a bi-matrix that displays the rewards for all combinations of actions. The first player chooses a row and the second player chooses a column. The numbers refer to the payoff for the first and second player respectively. If both players receive identical payoffs a simple matrix suffices.

The reward assigned to a player depends only on the current combination of actions. Multi-state games may introduce dependence on previous actions but are not within the scope of this paper. Furthermore, all games are deterministic, that is a unique payoff is assigned to each pure strategy profile.

The example in Figure 1 shows the payoffs in the Prisoners' Dilemma (PD), the classical text book example of a normal form game. It models the following strategic conflict: Assume two burglars are captured close to a crime scene and interviewed separately by the police. They can both choose to either confess or deny. If both confess, they will be sentenced for some years, if both deny they will be sentenced for one year for illegal possession of a weapon. However, if one confesses while the other one denies he is set free while the other one is sentenced for many years. The preference over the outcomes is displayed in the payoff value which is higher for more preferable outcomes.

Mathematically, an n-player normal form game is defined as the tuple $< N, (A_1, \dots, A_n), (u_1, \dots, u_n) >$:

- $N = \{1, \dots, n\}$ is a finite set of n players
- $A_i = \{a_{i1}, \dots, a_{im_i}\}$ is a finite set of m_i actions available to player i. The number of actions may differ between players. Let s_i take on the value of a particular action a_{ij} for player i. A pure strategy profile is an n-tuple $s = (s_1, \dots, s_n)$ that associates one action with each player. Furthermore, let $s_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$ denote this same profile without the action of player i, so that (s_i, s_{-i}) forms a complete profile of strategies.
- $u_i : A_1 \times \ldots \times A_n \rightarrow \Re$ is the utility function for

[1] Previous contributions can be counted to this discipline but this book triggered the first hype around game theory such that it was actually applied in practice [1].

player i. It maps a pure strategy profile to a value. For convenience the short hand notation $u_i(s_i|s_{-i})$ will be used besides $u_i(s)$ to refer to the utility of a complete profile of strategies.

Each player executes a pure strategy each iteration. It is drawn from the policy of player i, which is the mixed strategy π_i that assigns a probability to each action:

$$\pi_i \in \left\{ \pi_i : A_i \to [0,1] \mid \sum_{a_{ij} \in A_i} \pi_i(a_{ij}) = 1 \right\}$$

A mixed strategy π_i for player i specifies a probability distribution that is used to select the action $s_i = a_{ij}$ with probability $\pi_i(a_{ij})$ when the player plays the game. Let $\pi = (\pi_1, \ldots, \pi_n)$ denote the mixed strategy profile and $\pi_{-i} = (\pi_1, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_n)$ the same profile without the strategy for player i. The expected payoff for playing π_i against the set of mixed strategies π_{-i} played by the opponents is the sum over the utilities of all possible action combinations multiplied by their probability:

$$v_i(\pi_i|\pi_{-i}) = \sum_{s_i, s_{-i}} u_i(s_i|s_{-i}) \cdot \pi_i(s_i) \cdot \pi_{-i}(s_{-i})$$

The term $\pi_i(s_i)$ denotes the probability for player i to play the pure strategy s_i. The subsequent term $\pi_{-i}(s_{-i}) = \prod_{s_k \in s_{-i}} \pi_k(s_k)$ equals the probability that the other players play the strategy profile s_{-i}. It is easiest to grasp for a finite number of players. For two players the probability for each action combination is simply the product of two probabilities:

$$v_1(\pi_1|\pi_2) = \sum_{s_1, s_2} u_i(s_1|s_2) \cdot \pi_1(s_1) \cdot \pi_2(s_2)$$

2.2 Analyzing Games

A game has n players and m_i actions for player i. Two-agent games are often labeled with the number of actions for each player, e.g. a 2x2 game refers to a two-player game with two actions for both players. Some notions from GT aid in a more specific characterization.

Dominated Strategies

A pure strategy s_i is strictly dominated by s'_i if s'_i yields higher utilities for all opponents profiles:

$$\forall s_{-i} \; (u_i(s_i|s_{-i}) < u_i(s'_i|s_{-i}))$$

In the Prisoners' dilemma given in Figure 3 the pure strategy deny is strictly dominated by confess. A strategy is weakly dominated if $\forall s_{-i} \; (u_i(s_i|s_{-i}) \leq u_i(s'_i|s_{-i}))$.

Best Response

The best response is the set of strategies that have the maximal possible reward given all other players' strategies. Due to rationality all players are assumed to pick the best action available to them. A mixed strategy π is a best response of player i if there is no other mixed strategy π' that would lead to a higher reward for this player given that all other players' strategies π_{-i} remain the same.

$$BR(\pi_{-i}) = \pi_i \leftrightarrow \forall \pi'_i(\; v_i(\pi_i|\pi_{-i}) \geq v_i(\pi'_i|\pi_{-i}) \;)$$

Nash Equilibria

A Nash Equilibrium (NE) is a strategy profile for which no player can improve his payoff by changing his policy as long as the other players keep their policies fixed. It is a tuple of strategies $(\pi^*_1, \ldots, \pi^*_n)$ such that no player has an incentive for unilateral deviation, that is every strategy π^*_i is a best response to π^*_{-i}.

$$\pi^*_i = \arg\max_{\pi_i} v_i(\pi_i|\pi^*_{-i})$$

Figure 3 indicates all pure Nash equilibria by an asterisk.

Pareto Dominance

A strategy profile π Pareto dominates π' if and only if all players obtain at least the same reward and at least one player receives a strictly higher reward when π is played.

$$\begin{aligned} &\pi \textit{ Pareto dominates } \pi' \\ &\quad \leftrightarrow (\; \forall i (v_i(\pi) \geq v_i(\pi')) \land \exists j (v_j(\pi) > v_j(\pi')) \;) \end{aligned}$$

Pareto Optimality

A set of strategies is Pareto optimal if it is not Pareto dominated. It is important to notice that not every NE is Pareto optimal. The only NE (C, C) with utilities $(1, 1)$ in the PD for example is Pareto dominated by (D, D) with utilities $(3, 3)$.

Social Welfare

In a cooperative game it is desired to find a strategy profile that maximizes the overall outcome. The social welfare ω of a mixed strategy profile π is the sum of individual utilities:

$$\omega(\pi) = \sum_i v_i(\pi_i|\pi_{-i})$$

The actual social welfare of a pure strategy profile which is of interest in the context of reinforcement learning is defined as:

$$\omega(s) = \sum_i u_i(s_i|s_{-i})$$

In cooperative games it is desired to find a strategy profile that maximizes the social welfare, the maximum social welfare profile π^* is therefore defined as:

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

2.3 Selected Games

This section defines the games that serve as a benchmark for the learning algorithms under investigation. The most simple multi-agent games yield two players and two actions. Of those, three representative examples are selected. Two games with three actions are used to examine coordination problems between two agents with many actions. Finally, two games with many agents and many actions are laid out that reveal the ability of the learning algorithms to solve large scale problems.

Two-agent two-action games

Figure 2 defines the general payoffs for both agents in a game with two actions.

	L	R
T	a_{11}, b_{11}	a_{12}, b_{12}
B	a_{21}, b_{21}	a_{22}, b_{22}

Figure 2: General two-agent two-action game

Games of this type can be divided into three classes [14]:

Class 1 If $(a_{11} - a_{21})(a_{12} - a_{22}) > 0$ or $(b_{11} - b_{21})(b_{12} - b_{22}) > 0$ there exists at least one dominant strategy and therefore only one pure equilibrium. The only exception: Let player i have a dominant strategy s_i and the other player j obtain $u(s_j|s_i) = x \; \forall s_j$, then there are infinitely many equilibria where player j mixes arbitrarily between his actions.

Class 2 If $(a_{11} - a_{21})(a_{12} - a_{22}) < 0$, $(b_{11} - b_{21})(b_{12} - b_{22}) < 0$ and $(a_{11} - a_{21})(b_{11} - b_{12}) > 0$ there are two pure and one mixed equilibrium.

Class 3 If $(a_{11} - a_{21})(a_{12} - a_{22}) < 0$, $(b_{11} - b_{21})(b_{12} - b_{22}) < 0$ and $(a_{11} - a_{21})(b_{11} - b_{12}) < 0$ there is just one mixed equilibrium.

In each class of games the learning algorithms show different learning dynamics. This paper studies the examples given in Figure 3 where the classes are represented by the PD (class 1), the Battle of Sexes (BoS) (class 2) and the Matching Pennies (MP) (class 3).

The NE (C, C) of the PD is not Pareto optimal because it is Pareto dominated by the strategy pair (D, D). BoS is also known as Bach or Stravinsky. A couple wants to go out together but they need to decide for *Bach* or *Stravinsky* without communication. They have different preferences and only enjoy their evening if they meet their partner. It yields two pure equilibria at (B, B) with payoffs $(2, 1)$ and (S, S) with payoffs $(1, 2)$ and one mixed equilibrium where player 1 mixes between the actions $(\frac{2}{3}, \frac{1}{3})$ and player two mixes $(\frac{1}{3}, \frac{2}{3})$ which leads to expected payoffs $(\frac{2}{3}, \frac{2}{3})$. All equilibria are Pareto optimal but only the pure equilibria correspond to maximum social welfare profiles.

	D	C
D	$3, 3$	$0, 5$
C	$5, 0$	$1, 1^*$

	B	S
B	$2, 1^*$	$0, 0$
S	$0, 0$	$1, 2^*$

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

Figure 3: Reward matrices for Prisoners' Dilemma (top-left, *Deny* or *Confess*), Battle of Sexes (top-right, *Bach* or *Stravinski*) and Matching Pennies (bottom, *Head* or *Tail*); *=pure Nash equilibria

The MP game is played by two players revealing two coins simultaneously. Both can choose *Head* or *Tail*, player one wins if they reveal the same side of the coin, otherwise player two wins. Matching Pennies is also called the Parity Game because it is completely symmetric. In the mixed NE both players mix both actions equally and obtain expected rewards $(0, 0)$ which is Pareto optimal.

Penalty Games

Penalty games are a special type of coordination games. They feature payoffs that reward certain joint actions and punish mis-coordination. The Climbing Game (CG) and Penalty Game (PG) as defined in [2] are used in this paper.

This class of games can be divided into games with one and games with more than one optimal joint action. The CG has the single optimal joint action (T, L) while the PG yields two optimal joint actions at (T, L) and (B, R). As both classes are represented the scope can be restricted to these two games without loss of generality.

Climbing Game

The CG given in Figure 4 is a cooperative game with identical payoffs for both players. It has one optimal joint action (T, L) and Nash equilibria at (T, L) and (M, C). Furthermore, heavy penalties surround the Pareto optimal Nash equilibrium while the last action for each player is never penalized. This makes it a tough benchmark for independent learners that should overcome the penalties to find and reach the optimum. Common values for the penalty are $c = 30$ and $c = 10$.

Penalty Game

The PG exhibits the difficulty to agree on one optimal joint action, either (T, L) or (B, R) while encountering

	L	C	R
T	11^*	$-c$	0
M	$-c$	7^*	6
B	0	0	5

Figure 4: Payoff matrix for the climbing game, experiments use $c = 10$; *=pure Nash equilibria

heavy penalties for each mis-coordination. It is a cooperative game with identical payoffs given in Figure 5. The game yields three Nash equilibria, which are at (T, L), (M, C) and (B, R). The penalties and the Nash equilibrium in the center make it a challenging game for learners because the second action of each player guarantees a non-negative reward while the optima can only be achieved by cooperation. Common values for the penalty are $p = 0$ and $p = 10$.

	L	C	R
T	10^*	0	$-p$
M	0	2^*	0
B	$-p$	0	10^*

Figure 5: Payoff matrix for the penalty game, experiments use $p = 10$; *=pure Nash equilibria

Coordination Games

Games serve as a model for coordination tasks at least since 1960, when Schelling made the use of games in this context explicit by introducing the two-player guessing game [10].

Let the coordination game and the anti-coordination game be subsumed under the term coordination games. In the coordination game agents obtain a positive reward when they execute the same action, and in the anti-coordination game vice versa. The guessing game is an example of a coordination game.

In the two-agent two-action case given in Figure 6 the two classes of games are equivalent and can be converted from one to the other by renaming the actions. Once more than two agents or two actions are considered both games differentiate. In order to study the scalability of learners, the generalizations of these games to many agents and many actions should be investigated.

	A	B
A	1	0
B	0	1

	A	B
A	0	1
B	1	0

Figure 6: Two-agent two-action coordination game (left) and anti-coordination game (right).

Guessing Game

Imagine a situation where players take individual decisions but the individual payoff is proportional to the number of players that take the same action. The introduction of new standards in industry is such a process. Each company has to decide which standard to support and their utility depends on the number of other players that make the same choice. It is a generalization of the coordination game to arbitrary numbers of agents and actions and will be called Guessing Game (GG) in this article.

The reward for each agent could be given in a matrix but is easier to grasp as a formula. Let $g(s_i|s_{-i})$ be the number of agents choosing action s_i, that is the number of j's for which $s_j \in s$ and $s_j = s_i$. Recall s_i is the action player i chooses and n the total number of players. The utility function $u_i(s_i|s_{-i})$ for player i is defined as:

$$u_i(s_i|s_{-i}) = \frac{g(s_i|s_{-i})}{n} \tag{1}$$

The experiments are based on the complete guessing game which features as many actions as players, thus $m_i = n$. This version has n pure NE at $s = (s_1, \ldots, s_n)$ where $x \in \{1, \ldots, n\}$ and $s_i = a_{ix}$ for all i. Each of the NE is Pareto optimal with a utility value of $u_i(s_i|s_{-i}) = \frac{g(s_i|s_{-i})}{n} = \frac{n}{n} = 1$ for each agent.

Dispersion Game

A large number of natural problems, including load balancing in computer science, niche selection in economics and division of roles within a team in robotics, require agents to disperse as well as possible over a number of actions [5]. Any of these problems can be modeled by a Dispersion Game (DG) - a generalization of the anti-coordination game to arbitrary numbers of agents and actions.

Again, let $g(s_i|s_{-i})$ be defined as the number of agents choosing action s_i and recall s_i is the action player i chooses. The utility $u_i(s_i|s_{-i})$ for player i is one if no other player plays the same action and zero otherwise:

$$u_i(s_i|s_{-i}) = \begin{cases} 1 & if\ g(s_i|s_{-i}) = 1 \\ 0 & otherwise \end{cases} \tag{2}$$

Experiments are restricted to the complete dispersion game in which n actions are available to all players. This game has $n!$ optimal joint actions with a utility value of 1 for each player where each player plays a different action. The set of optimal joint actions equals the set of Nash equilibria which are all Pareto optimal.

3 Reinforcement Learning

Reinforcement Learning (RL) has originally been studied in the context of single-agent environments. An agent receives a numerical reward signal, which it seeks to maximize. The environment provides this signal as a feedback on the sequence of actions that has been executed by the agent. Figure 7 depicts the environment-agent interaction schematically. Learners relate the reward signal to previously executed actions to learn a policy that maximizes the expected future reward [12].

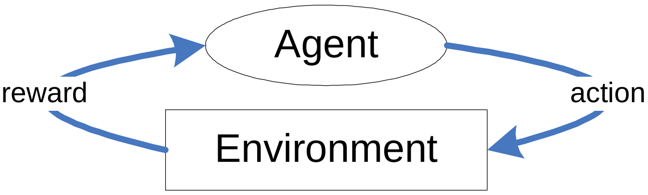

Figure 7: Agent-environment interaction, a feedback is given as a response to each action and may depend on the complete sequence of executed actions.

3.1 Q-learning

Q-learning was initially discussed for single-agent environments with states. Each learning step refines a utility-estimation function for state-action pairs and generates a new policy from the estimated values to draw the next action to execute. Q-learning with an ϵ-greedy action selection with $\epsilon = 0$ over Q-values has been proved to converge to the optimal policy under additional assumptions like an appropriate environment [16]. However, different action selection methods may outperform ϵ-greedy action selection in practice.

Figure 8 shows an example environment in which the agent needs to reach some goal state, in the example the mouse needs to find the cheese.

The idea originates from the Bellman optimality equation which is given in Equation 3. Let $R(p)$ be the reward for being in state p, $R(p) = 0$ for all states p except the goal state p^* for which $R(p^*) > 0$. $P(p'|p,a)$ denotes the probability to be in state p' given that action a is executed in state p and $\gamma \in [0,1]$ is the discount factor for future rewards. V^* estimates the value of a state by taking into account the immediate reward and the discounted, expected future rewards.

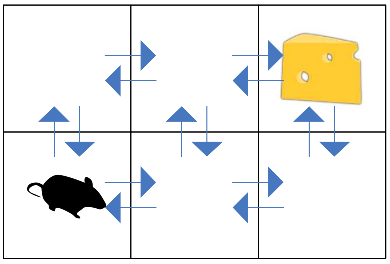

Figure 8: A grid world for which the state action pair values can be learned by the Q-learning algorithm. The mouse needs to select an action to get to the cheese.

$$V^*(p) = R(p) + \max_a \gamma \cdot \sum_{p'} P(p'|p,a) V^*(p') \qquad (3)$$

The optimal action a is given by

$$a = \arg\max_{a'} \gamma \cdot \sum_{p'} P(p'|p,a') V^*(p')$$

Q-learning leverages the state value estimation to relate rewards to state-action pairs.

Q-learning with states

Originally, Q-learning was discussed for a single agent leading to Equation 4 as described in [12]. Let $Q^t(p,a)$ be the quality estimation in iteration t of action a executed in state p. Let V be the state value estimation function that assigns the estimated value of the best available action in that state:

$$V^t(p) = \max_{a'} Q^t(p,a')$$

For the following formula from single agent learning, a is the action for which the reward r is obtained. Let p be the state in which action a is performed and let p' be the current state that is reached by playing a.

$$Q^t(p,a) \leftarrow (1-\alpha) \cdot Q^{t-1}(p,a) + \alpha \cdot (\, r + \gamma \cdot V^{t-1}(p') \,) \qquad (4)$$

The revised estimation is the weighted average between the current estimation and the sum of observed reward and expected discounted future rewards. An agent with this update rule can learn the Q-values for all actions in the grid world from Figure 8 and generate the optimal policy from them.

Multi-agent Q-learning without states

The games under consideration do not feature states hence the Q-function plainly estimates utilities of the available actions. Furthermore, each agent has an independent Q-value estimation function. Equation 5 shows the Q-update rule for stateless Q-learning using the following terms:

- $Q_i^t(s_i)$ Q-value estimation function of player i at iteration t for action s_i
- s_i^t Action of player i played in iteration t
- r_i^t Reward for player i obtained in iteration t
- $\alpha \in [0,1]$ Learning rate

The new estimation is the weighted sum of the old estimation and the observed reward.

$$Q_i^t(s_i) \leftarrow (1-\alpha) \cdot Q_i^{t-1}(s_i) + \alpha \cdot r_i^t \qquad (5)$$

As a result of (5) the Q-values of any iteration are bounded by the initial Q-values and the rewards an agent may encounter. This gains particular importance for the policy generation and restricts the domain of policies that can be learned. Therefore, initial Q-values should be chosen within the range of rewards because they would alter the learnable domain otherwise.

Both enhancements of Q-learning that are presented below are based on Equation 5. Besides the Q-update, the action selection method is crucial to the learner. Several approaches with different exploration-exploitation properties are explained in the next section.

3.2 Action Selection

In each iteration an action needs to be chosen based on the current knowledge. This step is essential to balance exploitation versus exploration. Three methods will be considered for this purpose: ϵ-greedy, normalization and the Boltzmann distribution.

Every action selection defines how to generate a policy from the value estimation. The policy is a distribution that is used to draw the action for the next iteration. Let the probabilities $p_j \in [0,1]$ for each action j be defined in the probability distribution vector $p = (p_1, \ldots, p_m) = \pi_i$ that defines policy π_i for player i. The elements of a valid distribution vector sum up to one:

$$\sum_j p_j = 1$$

An ϵ-greedy action selection assigns the best action with probability $(1-\epsilon)$ and some random action with probability ϵ. Let $\epsilon \in [0,1]$ be the exploration factor and let m be the number of actions.

$$p_j = \begin{cases} 1-\epsilon & if\ j = \arg\max\limits_{s_i} Q_i^t(s_i) \\ \frac{\epsilon}{m-1} & otherwise \end{cases}$$

This approach allows to alter the exploration-exploitation trade-off but has a primary emphasis on exploitation. Exploration is random and not directed toward promising alternatives.

The simplest method to generate a probability distribution that takes into account the balance between all q values is normalization. However, this requires non-negative Q-values and at least one strictly positive Q-value. Due to the fact that these values are bounded by the learning algorithm this translates into a restriction of non-negative rewards.

$$p_j = \frac{Q_i(s_j)}{\sum\limits_k Q_i(s_k)}$$

	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	-	-

Figure 9: 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 of the updated action (right).

However, normalization is not commonly applied in practice because it does not lead to convergence for all rewards as the learnable policies are very confined.

A dynamic trade-off between exploration and exploitation can be implemented using the idea of temperature from physics. The Boltzmann distribution allows a probability generation from arbitrary parameters. In contrast to normalization it relaxes the assumptions of positive q values. This approach is also often used for simulated annealing where an initially high temperature promotes exploration and decreasing temperature over time leads to strong exploitation in the final phase.

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

The Boltzmann distribution combines the advantages of the previous two methods. By tuning the temperature parameter τ the balance between exploration and exploitation can be adjusted while exploration is still directed toward promising actions. Very high values, e.g. $\tau = 500$, make the exploration random while very low values, e.g. $\tau = 0.01$, equal a greedy approach[2]. For $\tau = 1$ this equation is also called Gibbs distribution which is often applied for its intermediate behavior.

3.3 Lenient Q-learning

In a cooperative learning environment it might be good to forgive mistakes, especially in the initial learning period. Consider the example of learning in soccer as in [8]. In the initial phase of learning both agents lack the skill for good actions hence even a perfect forward pass may frequently be not rewarded. This lets the agents converge to actions that work well with a variety of opponents strategies but that often result in suboptimal strategy profiles. In order to tackle this problem lenient Q-learning collects L rewards for an action before it updates the estimation based on the maximum. Figure 9 depicts the update schematically. Lower rewards are discarded and only the highest reward is used for the update. This implies that only $\frac{1}{L} \cdot iterations$ learning steps are executed.

[2] Of course the given examples only apply in the context of the rewards that are used which bound the Q-values.

3.4 FMQ-Learning

The Frequency Maximum Q value (FMQ) - learner has emerged from the optimistic assumption [7]. It keeps track of the highest reward for each action and its frequency so far. This 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 shift the policy. This works best in combination with another FMQ learner such 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 a pure strategy.

4 Learning Behavior Analysis

This section presents the means by which learning behavior is studied [3].

Policy Trajectories

Trajectory plots display the learning path of I iterations in the policy space. Any policy for $n \times n$ games can be described by the probabilities of $n-1$ actions. Mixed strategy profile trajectories for 2x2 games can be plotted into $[0,1] \times [0,1]$ by plotting $(\pi_1^t(a_{11}), \pi_2^t(a_{21}))$ for all t. Mixed strategy profiles for 3x3 games would require a 4-dimensional space. Therefore, policies for 3x3 games are plotted separately for each player into standard 2-simplex plots. Each corner $c_j = (x_j, y_j)$ of the simplex represents the pure strategy s_j. The position of a policy in the simplex is the weighted average of the corner points.

$$position(\pi_i^t) = \sum_{j=1}^{m_i} \pi_i^t(s_j) \cdot c_j$$

Cyclic behavior, smoothness and stability become obvious in policy trajectory plots. Using a color-map even convergence speed can be visualized intuitively. The initial policies are displayed gray and the trajectory fades to the final policy which is black.

Directional Field Plots

Directional field plots capture the local dynamics of the learning behavior. They supply an overview over basins of attraction by displaying learning behavior at a set of grid points. The learner is started at a set of regular grid points in the interval $]0,1[$ with a specified granularity δp for the initial policy. After I iterations the distance between the initial policy profile π^0 and the final policy profile π^I defines the direction and length of the displayed arrow starting at the grid point that corresponds to π^0. It is averaged over R runs to average out stochastic influences.

Convergence

The analysis of convergence behavior requires a distance metric. When the distance from a policy profile π^t to π' is less than the threshold ϵ, π^t is considered converged to π'. The distance of a mixed strategy π_i to another mixed strategy π_i' is defined as:

$$d_i(\pi_i, \pi_i') = \max_j |\pi_i(s_j) - \pi_i'(s_j)|$$

The distance of a strategy profile π to another strategy profile π' is consequently defined as:

$$d(\pi, \pi') = \max_i d_i(\pi_i, \pi_i')$$

A strategy profile π is considered ϵ-near converged to π' at iteration T for some ϵ if the distance between the two profiles is less than ϵ from iteration T onwards:

$$\begin{aligned} &\pi \textit{ converged to } \pi' \textit{ at } T \\ &\quad \leftrightarrow \ \forall t \ (t \geq T \rightarrow d(\pi^t, \pi') < \epsilon) \end{aligned}$$

This definition is chosen because it has an intuitive interpretation. The strategy profile π is considered converged to π' at iteration T if no player plays any action with a probability that deviates more than ϵ from π in any later iteration of the game. The interpretation does not change when the number of actions or agents is altered.

All games under consideration are played a finite number of T_{max} iterations. A confidence interval for the mean convergence time $\tilde{T}$ can be created using N samples that average over M runs each. As long as M is chosen large enough the law of large numbers assures that the samples approximately follow a t-distribution with N degrees of freedom. The confidence interval can be used to determine if T_{max} is chosen large enough. Furthermore, it allows to detect premature convergence under bad parameter settings. The convergence time refers to the time when the learning process converged to *some* policy, which is not necessarily a NE.

Besides the temporal analysis, interesting points of convergence π^* can be studied. This provides insight into the convergence behavior to equilibria and tendencies toward local or global optima, e.g. in penalty games. All given intervals refer to 95% confidence intervals for the mean convergence behavior.

[3] The theoretical content of this section was enriched by mutual collaboration with the author of [6].

Social Welfare Plot

Cooperative games require the agents to maximize the overall outcome, the social welfare ω. Scaling experiments primarily investigate the relation between the number of agents and convergence speed to desired strategy profiles. The social welfare plot shows the development of the summed utilities of all agents over the iterations. It plots the percentage of the maximal social welfare $\frac{\omega(s^t)}{max_s \omega(s)}$ over iterations t. The normalization to the percentage makes different numbers of agents in the DG and GG comparable.

5 Experiments

This section presents the results obtained in the experiments and indicates parameter settings for which desired learning behavior can be achieved. The analysis of learning behaviors is centered around the regular Q-learner that is the basis for the two enhancements FMQ and lenient Q-learning. The enhancements will be compared where appropriate.

An independent learner will need to converge to the best reply in order to maximize his payoff. In self-play a good learner is expected to converge to the Nash equilibria as both players converge to a best reply against each other.

5.1 Performance in Simple Games

Each learner has different strengths and weaknesses so they are not suited equally well for each game. Even simple 2x2 games may require a good selection of the algorithm to apply and fine tuning of its parameters.

The regular Q-learner performs quite well but does only converges to mixed equilibria under certain temperatures. Several extensions have been devised to overcome this problem, e.g. extended replicator dynamics [13]. Table 1 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.

Prisoners' Dilemma (PD)

The Prisoners' dilemma yields one Nash equilibrium that is not Pareto optimal. Figure 10 visualizes the learning behavior of the regular Q-learner in the PD. Convergence remains bounded as discussed in Appendix A and the point of attraction can be moved toward the equilibrium by decreasing the temperature τ.

Table 1 shows convergence of FMQ and lenient Q to the Nash equilibrium. 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. Figure 11 shows example trajectories for both learners. It visualizes the information from Table 1, FMQ converges with a mean of approximately $I = 5$ iterations while lenient Q-learning requires about $I = 186$ iterations.

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

FMQ-learner $F = 3$

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

Lenient Q-learner $L = 3$

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

Matching Pennies (MP)

The MP game yields one mixed NE where both players mix both actions equally. Figure 12 visualizes the learning behavior of the three learners in the MP.

The FMQ learner with $F = 3$ converges to the mixed equilibrium most quickly, followed by the regular Q-learner. However, each learning step of the lenient

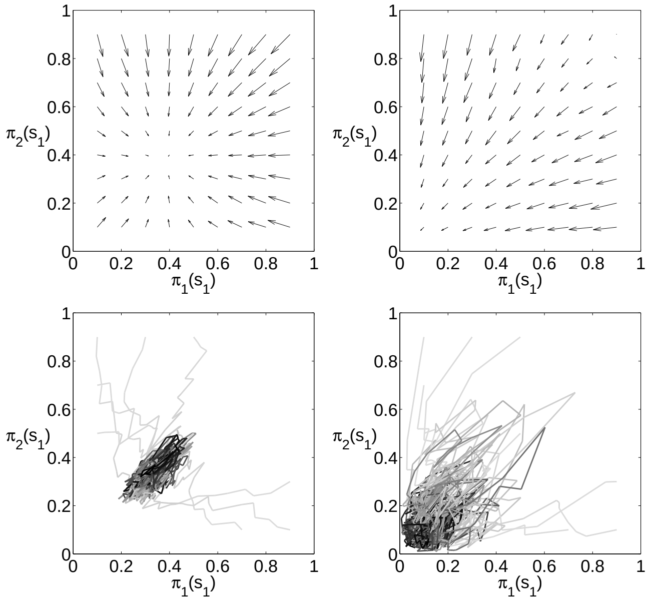

Figure 10: Directional field plots (top, $I = 200$) and example trajectories (bottom, $I = 600$) for Q-learner in PD under $\alpha = 0.1$, $\tau = 2$ (left) and $\tau = 0.5$ (right). The attractor close to the equilibrium can be analytically predicted as derived in Appendix A.

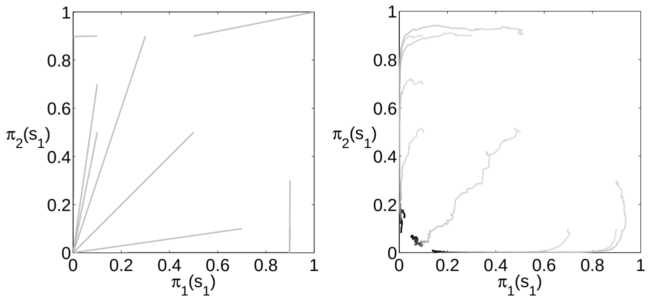

Figure 11: Example trajectories of $I = 2000$ iterations for FMQ (left, $F = 10$) and lenient Q-learner (right, $L = 3$) in PD under $\tau = 0.5$ and $\alpha = 0.01$. FMQ converges quickly while lenient Q-learning converges slowly. One trajectory converges to the Pareto optimal strategy profile (D, D).

Q-learner is more directed to the equilibrium than the Q-learner and convergence is only slowed down because of the reduced number of learning steps. If the persistence F for the FMQ learner is increased further, e.g. to $F = 10$, it does not find the mixed equilibrium anymore. In contrast to that the lenient Q-learner is very robust to parameter changes as long as the learning rate is small enough. Under $\alpha = 0.01$ the parameters $L \in \{3, \ldots, 10\}$ and $\tau \in [0.01, 1]$ lead to convergence to the mixed equilibrium.

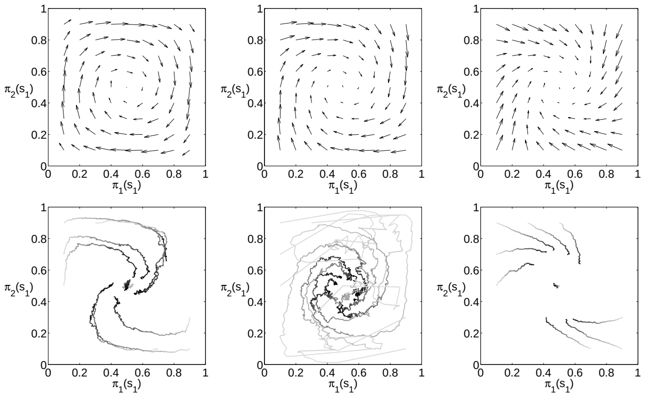

Figure 12: Directional field plots (top row, $I = 10$) and trajectories (bottom row, $I = 600$) in the MP for the Q-learner (left), FMQ ($F = 3$, center) and lenient Q-learner ($L = 3$, right) under $\alpha = 0.01$ and $\tau = 1$.

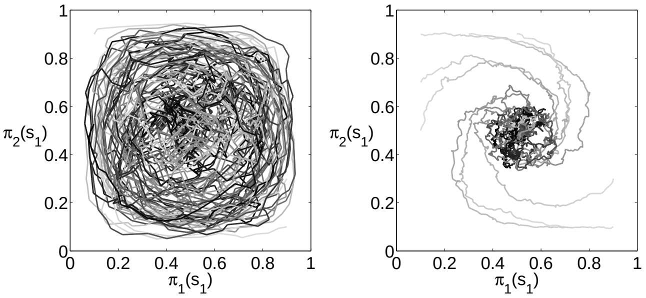

Figure 13: Trajectories ($I = 1000$) for the Q-learner in the MP under $\alpha = 0.01$ and $\tau = 0.1$ (left, no convergence to the equilibrium) and $\tau = 0.5$ (right, ϵ-near convergence to the equilibrium for $\epsilon = 0.1$). Learners circle around the equilibrium but with an appropriate temperature stay very close.

Figure 13 shows example trajectories of Q-learning in the MP game. It can be observed that the mixed equilibrium is not a stable point of convergence under $\tau = 0.1$ while it is under $\tau = 0.5$. Strong exploitation caused by small temperatures lead to a policy generation similar to ϵ-greedy. This forces large shifts in the policy and decreases the likelihood of convergence to mixed equilibria.

Battle of Sexes (BoS)

The game Battle of Sexes yields one mixed and two pure Nash equilibria. Figure 14 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 $\tau = 0.5$. The regular Q-learner converges to different equilibria dependent on the temperature τ as shown in Figure 15. However, the mixed equilibrium is instable and after sufficient iterations all trajectories converge to a pure NE.

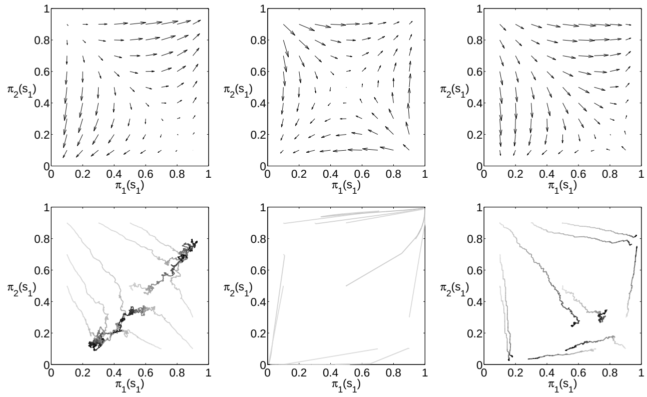

Figure 14: Directional field plots ($I = 20$, top row) and example trajectories (bottom row) for Q-learner (left), FMQ ($F = 3$, center) and lenient Q-learning ($L = 3$, right) in BoS under $\tau = 0.5$ and $\alpha = 0.01$.

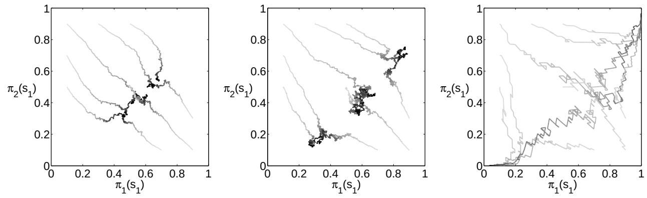

Figure 15: Q-learner trajectories in BoS under $\alpha = 0.01$ and $\tau = 1$ (left), $\tau = 0.5$ (center) and $\tau = 0.1$ (right); higher temperatures lead to convergence to the mixed equilibrium but bound convergence possibility to pure equilibria while low temperatures enforce fast convergence to pure equilibria only.

5.2 Performance in Penalty Games

Penalty games are cooperative games with high penalties around the desired maximum social welfare profile π^*. The regular Q-learner does not overcome the penalties systematically which initiated the development of the two adaptations under investigation: FMQ- and lenient Q-learning.

The results for regular Q-learning and FMQ learning from [7] as well as the results for lenient Q-learning from [9] are confirmed. Table 2 compares the algorithms' performances in both games and summarizes the behavior that tables 3 and 4 describe in detail. All results of this section 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 $\tau^0 = 500$.

$$\tau^t \leftarrow (\tau^0 - 1) \cdot e^{-s \cdot t} + 1 \qquad (6)$$

Table 2: 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%

Climbing Game (CG)

Table 3 lists the confidence intervals of ϵ-near convergence in the climbing game with $\epsilon = 0.1$ to pure strategy profiles in percentages. Example trajectories of the FMQ-learner are visualized in Figure 16.

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 hence it is as well the best strategy profile for independent learners.

The climbing game can not be solved satisfactory by the regular Q-learner. Q-learning converges to π^* in about 21.8% while both adaptations outperform this by far. FMQ ($F = 10$) achieves 98.9% and lenient Q-learning ($L = 10$) 99.9% convergence to π^*.

Penalty Game (PG)

The penalty game as given in Figure 5 yields two Pareto optimal Nash equilibria at (T, L) and (B, R) that are maximal social welfare profiles. Table 4 list the confidence intervals of ϵ-near convergence with $\epsilon = 0.1$ to pure strategy profiles in percentages. The regular Q-learner converges to one of the desired policies in 79.6% while FMQ with $F = 10$ achieves 100% and lenient Q-learning with $L = 5$ achieves 99.3%.

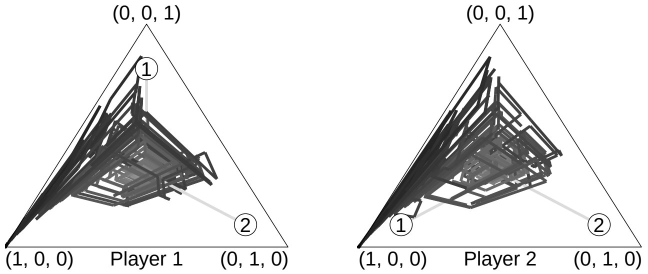

Figure 16: Two example trajectories for both FMQ-learners with $F = 3$ in the CG show convergence to the global optimum (T, L) starting close to (B, L) in (1) and (M, C) in (2). Initial high exploration causes large policy shifts while eventual exploitation allows convergence.

Table 3: 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

Table 4: 95% Confidence intervals for ϵ-near convergence with $\epsilon = 0.1$ in PG to pure strategy profiles in percent. Analyzed after $I = 2000$ iterations with $\alpha = 0.9$ and decreasing τ. Q-learner (top, 14.8% not converged to any pure strategy profile), FMQ-learner (middle, $F = 10$, 0.0% n.c.) and lenient Q-learner (bottom, $L = 5$, 0.7% n.c.). Player 1 chooses T, M or B and player two chooses L, C or R. (T, L) and (B, R) are maximal social welfare profiles.

Q-learner

	L	C	R
T	40.9 ± 2.2	0.0 ± 0.0	0.0 ± 0.0
M	0.0 ± 0.0	5.6 ± 1.0	0.0 ± 0.0
B	0.0 ± 0.0	0.0 ± 0.0	38.7 ± 2.3

FMQ-learner $F = 10$

	L	C	R
T	50.0 ± 2.4	0.0 ± 0.0	0.0 ± 0.0
M	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
B	0.0 ± 0.0	0.0 ± 0.0	50.0 ± 2.4

Lenient Q-learner $L = 5$

	L	C	R
T	49.9 ± 2.2	0.0 ± 0.0	0.0 ± 0.0
M	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
B	0.0 ± 0.0	0.0 ± 0.0	49.4 ± 2.2

5.3 Scalability

Many real life problems involve large numbers of agents. In order to give an impression how well the algorithms can be applied to large scale problems they are tested in the GG and the DG with varying numbers of agents. Scaling experiments require specific parameter values and a wrong choice easily leads to non converging behavior. 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.

Q-values are initialized with the average of the minimal and maximal reward to encounter, $Q_i^0(s_i) = \frac{1}{2}$ for all players i and all actions s_i.

Guessing Game (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 17 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 increase the qualitative convergence while lenience slows down the learning process but allows convergence to a maximal social welfare equilibrium.

Dispersion Game (DG)

Figure 18 visualizes the impact of the policy generation method on the speed of dispersion in the DG. 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, even for $n = 1000$ a performance of 90% is reached within 50 iterations.

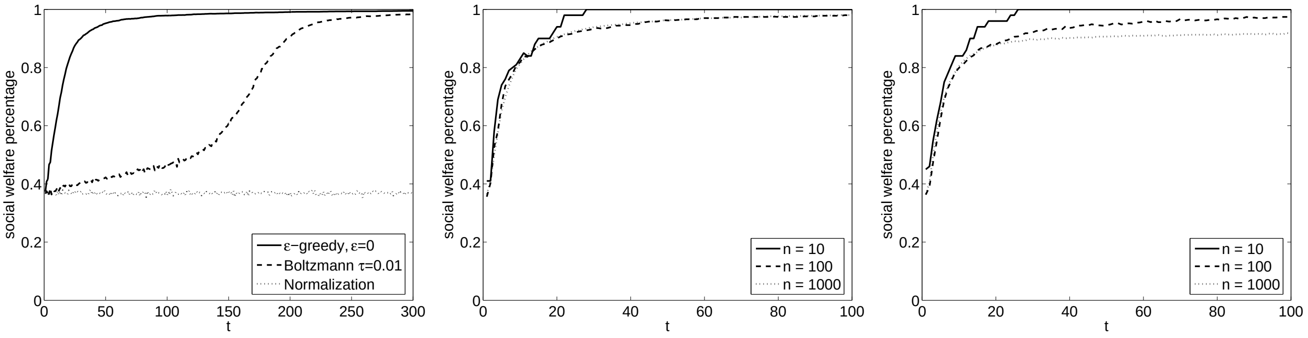

Figure 18: Social welfare percentage over iterations in the DG (averages of 10 runs). Q-learner under $\alpha = 0.1$, different action selection methods (left, $n = 1000$), different numbers of agents: Q-learner with ϵ-greedy action selection (center, $\epsilon = 0$) and FMQ with Boltzmann distribution (right, $F = 10$, $\tau = 0.01$).

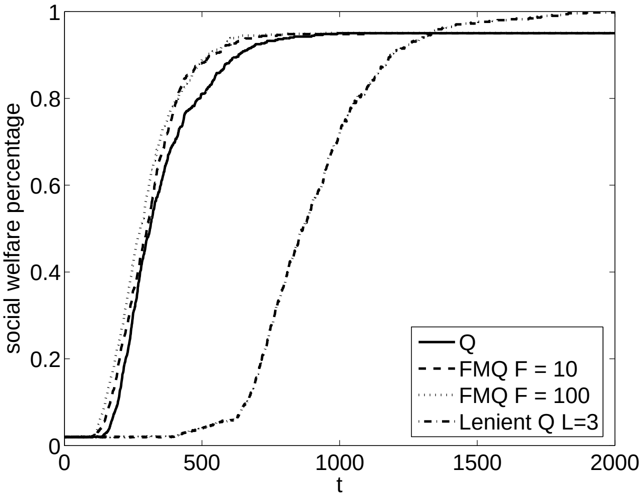

Figure 17: Social welfare percentage over iterations in the GG for different learners ($n = 100$, $\alpha = 1$ and $\tau = 0.01$; averaged over 10 runs). All learners except the lenient Q-learner converged to the suboptimal solution with two equally sized groups once.

6 Discussion

The obtained results demonstrate the learning performance of Q-learning in comparison with the adaptations FMQ and lenient Q-learning. Parameter settings for convergence to equilibria have been pointed out. In general, low temperatures and accompanying high exploitation lead to convergence to pure strategy profiles while higher temperatures that impose more exploration allow converge to mixed equilibria. Furthermore, higher convergence to mixed equilibria is achieved with smaller learning rates. On the other hand, 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.

Overall it can be observed that all results are very sensitive to the parameter settings. Due to the limited time not all parameter combinations could be taken into account, as a matter of that some algorithms may perform better on some games given different parameters.

Furthermore, simulation and initialization artifacts as discussed in Appendix A may have biased some results. The non-convergence to Nash equilibria in simple games, e.g. FMQ or lenient Q-learning in the PD, may be such a result. However, all results are valid under the given assumptions like the initialization and number of iterations to use. The intricate impact of simulation analysis stresses the need for analytical means to study learning behavior. Replicator dynamics and other means from evolutionary game theory have been successfully applied for this purpose [14].

Scaling experiments in the DG 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.

The GG 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, conducted experiments need to be repeated with higher number of runs before final conclusions can be drawn. An interesting opportunity of future work is also the investigation of mixed groups of learners. It could be investigated if FMQ learners initiate quicker grouping in heterogeneous groups. This article can serve as a basis for future work investigating the behavior of Q-learners in multi-state or stochastic games.

7 Conclusions

The performance of independent reinforcement learners has been shown to be highly dependent on the correct parameter tuning. In general, 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.

The contributions of this paper can be summarized as follows: Stateless Q-learning and the two adaptations Frequency Maximum Q-value (FMQ) and lenient Q-learning have been compared in games from game theory. Parameter settings that lead to convergence to Nash equilibria and the mean time for ϵ-near convergence are given. Furthermore, Q-learning has been shown to scale well with an ϵ-greedy action selection comparable with FMQ learning using the Boltzmann distribution for policy generation. Simulation analysis and visualizations have promoted a better understanding of learning dynamics of value iterators in single-state multi-agent games.

References

[1] Binmore, K. (1992). *Fun and Games.* D. C. Heath and Company.

[2] Claus, Caroline and Boutilier, Craig (1998). The dynamics of reinforcement learning in cooperative multiagent systems. *Proceedings of the National Conference on Artificial Intelligence*, Vol. 15, pp. 746–752.

[3] Gibbons, R. (1992). *A Primer in Game Theory.* Harvester Wheatsheaf.

[4] Gintis, H. (2000). *Game Theory Evolving.* Princeton University Press.

[5] Grenager, T., Powers, R., and Shoham, Y. (2002). Dispersion games: General definitions and some specific learning results. *Eighteenth National Conference on Artificial intelligence*, pp. 398–403.

[6] Hennes, D. (2007). Reinforcement learning in multi-agent games - A policy based perspective. *BSc thesis, Universiteit Maastricht.*

[7] Kapetanakis, S. and Kudenko, D. (2002). Reinforcement learning of coordination in cooperative multi-agent systems.

[8] Panait, L. and Tuyls, K. (2007). Theoretical advantages of lenient q-learners: An evolutionary game theoretic perspective. *AAMAS 2007.*

[9] Panait, Liviu, Sullivan, Keith, and Luke, Sean (2006). Lenience towards teammates helps in cooperative multiagent learning. *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi Agent Systems – AAMAS-2006*, ACM.

[10] Schelling, T. (1960). *The Strategy of Conflict.* Harvard University Press.

[11] Sen, S., Sekaran, M., and Hale, J. (1994). Learning to coordinate without sharing information. *Twelfth National Conference on Artificial Intelligence*, pp. 426–431.

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

[13] Tuyls, Karl, Heytens, Dries, Nowe, Ann, and Manderick, Bernard (2003). Extended replicator dynamics as a key to reinforcement learning in multi-agent systems. *Proceedings of the 14th European Conference on Machine Learning (ECML03), LNAI Volume: 2837*, Cavtat-Dubrovnik, Croatia.

[14] Tuyls, K., Hoen, P.J. t, and Vanschoenwinkel, B. (2005). An evolutionary dynamical analysis of multi-agent learning in iterated games. *Autonomous Agents and Multi-Agent Systems*, Vol. 12, pp. 115–153.

[15] Neumann, J. van and Morgenstern, O. (1944). *The Theory of Games and Economic Behavior.* Princeton University Press.

[16] Watkins (1989). Learning from delayed rewards. *PhD thesis, King's College, Oxford.*

A Inverted Boltzmann Distribution

A.1 Initialization of Q-values

The Boltzmann distribution can be inverted to generate initial Q-value estimations from the desired initial policy π^0. The values that Equation 7 defines as Q_i^{-1} ensure the correct balance such that the desired policy is generated by the Boltzmann distribution.

$$Q_i^{-1}(a_j) \leftarrow \tau \cdot \log \pi_i^0(a_j) \tag{7}$$

The raw Q-values given by Q_i^{-1} would not necessarily be achievable by the regular learning process. Once the Q-values are within the range of the minimal and maximal utility that the player may encounter they stay within that bound. A linear transformation into the valid Q-value space is applied to facilitate a realistic development from the initial Q-values. Let $L_i = \min_s u_i(s)$ and $U_i = \max_s u_i(s)$ denote the minimal and maximal utility value that player i may encounter. Equation 8 shifts the Q-values such that they are centered with respect to the valid space of Q-values for the corresponding player.

$$\begin{aligned} Q_i^0(a_j) \leftarrow & \quad Q_i^{-1}(a_j) \\ & -\tfrac{1}{2} \cdot (\min_k Q_i^{-1}(a_k) + \max_k Q_i^{-1}(a_k)) \\ & +\tfrac{1}{2} \cdot (L_i + U_i)2 \end{aligned} \tag{8}$$

Equation 9 shows that a shift by an arbitrary number t cancels out in the Boltzmann distribution.

$$\begin{aligned} \frac{e^{(Q_i(s_j)+t)\cdot\tau^{-1}}}{\sum_k e^{(Q_i(s_k)+t)\cdot\tau^{-1}}} &= \frac{e^t \cdot e^{Q_i(s_j)\cdot\tau^{-1}}}{e^t \cdot \sum_k e^{Q_i(s_k)\cdot\tau^{-1}}} \\ &= \frac{e^{Q_i(s_j)\cdot\tau^{-1}}}{\sum_k e^{Q_i(s_k)\cdot\tau^{-1}}} = p_j \end{aligned} \tag{9}$$

It is important to notice that Equation 8 does not guarantee valid Q-values under all circumstances. If the range of payoffs $U_i - L_i$ is smaller than the range of raw Q-values Q_i^{-1} then the initial Q-values Q_i^0 will still exceed the regular space of Q-values which may result in irregular behavior of the Q-learner.

$$\begin{aligned} & U_i - L_i < \max_k Q_i^{-1}(a_k) - \min_k Q_i^{-1}(a_k) \\ & \quad \rightarrow (\ \exists l\ (L_i > Q_i^0(a_l)) \wedge \exists m\ (U_i < Q_i^0(a_m)))\ ) \end{aligned}$$

A.2 Bounded Convergence

Another result of the bounded learning is a limited convergence toward any pure strategy profile if the Boltzmann policy generation is applied. If all players are sufficiently converged to the pure strategy profile s, then the Q-value of player i for any strategy s_j approaches the utility for the strategy profile (s_j, s_{-i}):

$$\lim_{t\to\infty} Q_i^t(s_j) = u_i(s_j|s_{-i}) \tag{10}$$

Equation 10 predicts that Q-learning converges to any pure equilibrium with at most:

$$\pi_i(s_j) = \frac{e^{u_i(s_j|s_{-i})\cdot\tau^{-1}}}{\sum_k e^{u_i(s_k|s_{-i})\cdot\tau^{-1}}} \tag{11}$$

This implies the convergence is directly dependent on the payoffs and the temperature τ. For the Prisoners dilemma as given in Figure 3 the probability to play the equilibrium action s_1 is given in Equation 12.

$$\begin{aligned} \pi_i(s_1) &= \frac{e^{1\cdot\tau^{-1}}}{e^{1\cdot\tau^{-1}} + e^{0\cdot\tau^{-1}}} \\ &= \frac{e^{\tau^{-1}}}{e^{\tau^{-1}} + 1} \end{aligned} \tag{12}$$

This probability approaches one when τ is small, zero when τ is large and is approximately 0.731 for $\tau = 1$. Recall the assumption that all players are sufficiently converged, in case of a low probability the rewards to consider would change hence the probability zero will never be achieved.

B Acronyms

BoS Battle of Sexes

CG Climbing Game

CRDG Crisp Reward Dispersion Game

DFP Directional Field Plot

DG Dispersion Game

EGT Evolutionary Game Theory

ESS Evolutionary Stable Strategies

FMQ Frequency Maximum Q value

GG Guessing Game

GT Game Theory

MDPs Markov Decision Processes

MP Matching Pennies

NE Nash Equilibrium From Game Theory

PG Penalty Game

PD Prisoners' Dilemma

RL Reinforcement Learning