Lenient Frequency Adjusted Q-learning

Daan Bloembergen Michael Kaisers Karl Tuyls

Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands

Abstract

Overcoming convergence to suboptimal solutions in cooperative multi-agent games has been a main challenge in reinforcement learning. The concept of "leniency" has been proposed to be more forgiving for initial mis-coordination. It has been shown theoretically that an arbitrarily high certainty of convergence to the global optimum can be achieved by increasing the degree of leniency, but the relation of the evolutionary game theoretic model to the Lenient Q-learning algorithm relied on the simplifying assumption that all actions would be updated simultaneously. Building on insights from Frequency Adjusted Q-learning, this article introduces the variation Lenient Frequency Adjusted Q-learning that matches the theoretical model precisely, and allows for arbitrarily high convergence to Pareto optimal equilibria in cooperative games.

1 Introduction

Many strategic interactions can be characterized as collaborative, i.e., success primarily depends on the coordination of actions executed by different agents. Such cooperative multi-agent games may yield a vast number of stable but suboptimal solutions, in which either the agents are partitioned into coordinated subgroups, or complete coordination on suboptimal joint actions is achieved. Cooperative multi-agent learning aims to overcome these challenges. However, multi-agent learning is significantly more complex than single-agent learning, since the presence of other adaptive agents makes the environment dynamic, and the optimal behavior depends on the other agents' strategies.

Recently, an evolutionary game theoretic approach to multi-agent reinforcement learning has been proposed to facilitate the understanding of multi-agent learning dynamics [1, 10]. This approach replaces individual rationality from game theory by concepts like selection and mutation from evolutionary biology to describe the change in a population of candidate strategies. The replicator dynamics that govern the population change have been formally linked to the behavioral change of multi-agent reinforcement learners.

This article demonstrates the strength of the evolutionary game theoretic approach by introducing a variation of Q-learning based on insights from evolutionary game theory. Recently, it has been shown that the introduction of leniency to the evolutionary model of Q-learning improves convergence to Pareto optimal equilibria in cooperative games [8]. A lenient version of Frequency Adjusted Q-learning [5] is proposed, which implements this evolutionary model. In addition, the match between the model and the proposed learning algorithm is demonstrated empirically, and the almost certain convergence to Pareto optimal equilibria with growing degree of leniency is illustrated.

The remainder of this article is structured as follows. Section 2 summarizes the necessary background on evolutionary game theory and reinforcement learning. The formal link between these two fields is presented and the proposed Lenient Frequency Adjusted Q-learning algorithm is introduced subsequently in Section 3. The theoretical arguments are complemented in Section 4 by an empirical study of the new algorithm and its dynamics. These experiments are discussed in Section 5, which also concludes this article.

2 Background

This section presents a brief overview of evolutionary game theory and reinforcement learning. For a more elaborate discussion of these two fields, the interested reader is referred to [4] and [9], respectively.

2.1 Evolutionary game theory

Game theory models the interaction between players as a game, in which each player has a set of actions to choose from. All players have to select an action simultaneously, upon which they receive a payoff that depends on the combination of actions played. The goal for each player is to come up with a strategy that maximizes its payoff in the game. It is assumed that the players are rational, in the sense that each player tries to maximize its own payoff irrespective of the payoffs of the others [3].

The payoffs can be conveniently represented in the bi-matrix (A, B). A bi-matrix can, just like a normal matrix, contain any number of rows and columns; "bi" just reflects the fact that each cell contains two numbers: the payoffs for both players [3]. Suppose the row player plays action i and the column player plays j, then the bi-matrix (A, B) gives the payoffs A_{ij} to the row player and B_{ij} to the column player. Figure 1 presents the payoff bi-matrices of three games that are used to evaluate the proposed learning method in this article. The Coordination Game and the Stag Hunt are cooperative games, where both players prefer the same outcome: (S, S) in the Stag Hunt and (O, O) in the Coordination Game. Matching Pennies is an example of a competitive game, in which a win for one player is a loss for the other.

	O	F
O	$1, 1$	$0, 0$
F	$0, 0$	$\frac{1}{2}, \frac{1}{2}$

Coordination Game

	S	H
S	$1, 1$	$0, \frac{2}{3}$
H	$\frac{2}{3}, 0$	$\frac{2}{3}, \frac{2}{3}$

Stag Hunt

	H	T
H	$0, 1$	$1, 0$
T	$1, 0$	$0, 1$

Matching Pennies

Figure 1: Normalized payoff matrices for the three games considered in this article.

2.1.1 Pareto optimality

An important concept in game theory is the Nash equilibrium (NE). A set of strategies for all players forms a Nash equilibrium if no single player can do better by playing a different strategy [3]. In the Stag Hunt, both (S, S) and (H, H) are NE; the Coordination Game has NE (O, O) and (F, F); and Matching Pennies has a mixed NE in which both players play action H with probability $\frac{1}{2}$. The examples of the Stag Hunt and Coordination Game show that not all NE of a game may be equally preferable. The concept op Pareto optimality captures this idea: an outcome is Pareto optimal if no other outcome leads to a higher payoff for at least one payer, without reducing the payoff to any other player [4]. The outcomes (S, S) and (O, O) are the Pareto optimal NE of the Stag Hunt and the Coordination Game; Matching Pennies has only one NE which is therefore also Pareto optimal.

2.1.2 Replicator dynamics

Classical game theory assumes that full information is available to the player, which together with the assumption of hyper-rationality does not reflect the dynamical nature of most real world environments [4]. Evolutionary Game Theory (EGT) was developed to overcome this limitation, by adopting the idea of evolution from biology to describe how players can learn to optimize their strategy without requiring complete information [6]. The theory provides a solid basis to study the decision making process of boundedly rational players in an uncertain environment.

Central to evolutionary game theory are the replicator dynamics, that describe how a population of candidate strategies evolves over time. Supposing that each player is represented by a population consisting of pure strategies, the fact that a player plays action A with probability p can be translated as a fraction p of the population playing pure strategy A. In a 2-player game, the process of change over time in the frequency distribution of the candidate strategies is described by

$$\frac{dx_i}{dt} = x_i[(Ay)_i - x^T Ay] \tag{1}$$

$$\frac{dy_i}{dt} = y_i[(Bx)_i - y^T Bx] \tag{2}$$

where x (y) is the frequency distribution for player 1 (2), and A (B) represents its individual payoff matrix. These equations are the replicator dynamics that constitute the link between EGT and RL.

2.2 Reinforcement learning

A reinforcement learning agent has to learn by trial-and-error interaction with its environment. It has no explicit knowledge on how to achieve its goal, it can only perceive the results of its actions by means of punishment and reward. The agent's goal is to maximize its reward over time by learning what the best action is in each situation. This article considers single-state reinforcement learning. Each time step the agent performs an action i upon which it receives a reward $r_i \in [0, 1]$. Based on this reward the agent updates its policy which is defined as a probability distribution over its actions x, where x_i denotes the probability of selecting action i. The way in which the policy is updated is specific to each reinforcement learning algorithm.

A distinction has to be made between single-agent and multi-agent RL. Whereas in single-agent environments the Markov property may be assumed, multi-agent environments are inherently non-stationary, and as a result proofs of convergence that hold for the single-agent case are no longer valid. Nevertheless, single-agent RL algorithms have been shown empirically to produce good results in multi-agent settings as well [2]. This article considers a variation of the Q-learning algorithm, which is introduced below.

2.2.1 Q-learning

Q-learning is an off-policy reinforcement learning method based on the idea of temporal difference (TD) learning [12]. TD methods generally consist of two steps: policy evaluation and policy iteration. The first step estimates a value function Q, that is then used in the second step to update the policy x. Q-learning differs from on-policy TD methods in that it approximates the true action-value function Q^* independent of the policy being followed [9]. Single-state Q-learning uses the action-value update function

$$Q_i(t+1) \leftarrow Q_i(t) + \alpha \left[r_i(t+1) + \gamma \max_j Q_j(t) - Q_i(t) \right] \tag{3}$$

to refine Q at every time step, where i is the action taken at time t, α controls the learning step size, and γ discounts future rewards. Only the value of the selected action is updated; for all other actions j, $Q_j(t+1) \leftarrow Q_j(t)$. The policy plays no role in this update process; it is only used to determine which action is selected. Instead, the action-value update is based purely on the reward received and the expected value of the next iteration, expressed by the term $\gamma \max_j Q_j(t)$. After each update of Q, the new optimal policy is derived using the Boltzmann exploration mechanism that converts the action-value function Q to the probability distribution x, using a temperature parameter τ to control the balance between exploration and exploitation:

$$x_i = \frac{e^{Q_i \cdot \tau^{-1}}}{\sum_j e^{Q_j \cdot \tau^{-1}}} \tag{4}$$

A high temperature drives the mechanism towards exploration by leveling the action probabilities, whereas a low temperature promotes exploitation by favoring actions with a high Q-value. Q-learning is proven to converge to the true action-value function Q^* in a Markovian environment, given that each action is selected (and its action-value is updated) an infinite number of times [12].

2.2.2 Lenient Q-learning

Lenient Q-learning is a learning method specifically tailored to cooperative multi-agent environments. When multiple independent agents learn together in such an environment, it can often happen that they converge to suboptimal solutions whereas a single agent might have no difficulty at all in finding the optimum. One of the reasons is that the environment is unpredictable, since actions taken by other agents influence the rewards received. Initial mis-coordination on a Pareto optimal solution may result in severe punishment, and as a result the Q-value of the Pareto optimal action may decrease. In the end, this can drive the agents away from the Pareto optimum, resulting in suboptimal behavior. This effect can be reduced by introducing leniency, i.e., by ignoring initial mis-coordination. It has been shown that leniency can greatly improve the accuracy of an agent's projection of the search space in the beginning of the learning process [7]. It thereby overcomes the problem that initial mis-coordination might lead to suboptimal solutions in the long run.

Leniency towards others can be achieved by having the agent ignore some low rewards and only consider the highest of several samples. For example, the agent might always update its policy when an action yields a higher reward than expected. When the reward is lower, it chooses probabilistically: in the beginning it

should show more lenience towards its teammates and ignore more low rewards, whereas in the end it might be more critical and always update its policy. A simpler approach is to have the agent collect κ rewards for a single action before it updates the value of this action based on the highest of those κ rewards [8]. This results in a fixed degree of leniency, expressed by the value of κ.

3 Connecting evolutionary game theory and reinforcement learning

The relation between reinforcement learning and evolutionary game theory was first formalized by [1], who proved that the continuous time limit of Cross learning, a specific reinforcement learning method, converges to the replicator dynamics. Recently, evolutionary dynamical models of Q-learning and Lenient Q-learning have been developed with the intention to provide a more intuitive way to understand the behavior of reinforcement learning [11, 8]. However, these two models use the simplifying assumption that all actions would be updated at each iteration. This leads to a discrepancy between the predicted and actual learning behavior. For Q-learning, this difference has been resolved by the variation Frequency Adjusted Q-learning (FAQ), which inherits the preferable behavior of the evolutionary model [5]. Combining the insights from leniency and frequency adjustment, a lenient version of the FAQ algorithm, called Lenient Frequency Adjusted Q-learning (LFAQ), is proposed that adheres to the predictions of the evolutionary model of Lenient Q-learning in the same way that FAQ adheres to the model of Q-learning.

3.1 Frequency Adjusted Q-learning

Differences between the actual behavior of reinforcement learning and the evolutionary model have been observed. It is known that such differences can occur when the step size of the learning is large [1]. However, the discrepancies observed in Q-learning may remain when the step size is decreased [5]. The evolutionary model was derived under the assumption that all actions are updated equally often [11], but the action-values in Q-learning are updated asynchronously: the value of an action is only updated when it is selected. If synchronous updates incur a change of ΔQ_i, then asynchronous updates incur a change of $x_i \Delta Q_i$, where x_i is the probability of selecting action i. Due to the simplifying assumption, the derivative of the action-value update function used in the evolutionary model of Q-learning differs by a factor x_i from the actual Q-learning behavior. It is further argued that the evolutionary model predicts more rational behavior than the Q-learning algorithm actually exhibits, and therefore [5] introduce the variation Frequency Adjusted Q-learning (FAQ) that perfectly fits the evolutionary model. The action-value update is weighted inversely proportional to the probability with which the action is selected, thereby simulating synchronous updates:

$$Q_i(t+1) \leftarrow Q_i(t) + \frac{1}{x_i}\alpha \left[r(t+1) + \gamma \max_j Q_j(t) - Q_i(t) \right] \tag{5}$$

However, this function is only valid in the infinitesimal limit of α, as otherwise the fraction α / x_i may become larger than 1. This would violate the assumptions under which the algorithm converges [12]. In order for the method to be numerically applicable, the authors define a generalized version of the frequency adjusted Q-learning method, by introducing a variable $\beta \in [0, 1)$:

$$Q_i(t+1) \leftarrow Q_i(t) + \min\left(\frac{\beta}{x_i}, 1\right) \alpha \left[r(t+1) + \gamma \max_j Q_j(t) - Q_i(t) \right] \tag{6}$$

When $\beta = 1$, FAQ reduces to normal Q-learning; therefore this value is excluded from the allowed range of β. It is shown that the value of β controls the area of the policy space for which FAQ is valid [5]. If $x_i \geq \beta$, FAQ behaves according to the evolutionary dynamics; if $x_i < \beta$, FAQ behaves equivalent to regular Q-learning. Given the range of possible rewards and a specific temperature τ, the most extreme policy that may arise can be computed using the Boltzmann policy generating function. Hence, a temperature τ can be selected according to β, such that $x_i \geq \beta$ is guaranteed in FAQ-learning, and thus the algorithm always behaves according to the evolutionary model. Theoretically, this means that the range of β can be further narrowed, since it reduces the policy space from both sides: $0 < \beta < \frac{1}{2}$. Practically, β should be chosen as small as possible to ensure the validity of FAQ for a large part of the policy space.

The behavior of FAQ has been shown to match the evolutionary model that was originally designed for Q-learning, whereas Q-learning itself deviates from it [5]. Furthermore, FAQ is less sensitive to the initialization of the Q-values, whereas Q-learning behaves differently depending on the initial action-values. The latter fact makes FAQ a robust choice for many applications where correct initialization might be impossible.

3.2 Lenient Frequency Adjusted Q-learning

Several ways exist in which leniency can be introduced in a learning method, as explained in Section 2.2.2. The most straightforward way to forgive mistakes is to collect several rewards for each action, before performing an update step. This update is then based on the highest of those collected rewards. The expected maximum payoff for action a_i over κ interactions is given by Equations 7 and 8 for both players. Substituting these for the reward matrices A and B in the replicator dynamics leads to the evolutionary model of Lenient Q-learning [8]:

$$u_i = \sum_j \frac{A_{ij} y_j \left[\left(\sum_{k:A_{ik} \leq A_{ij}} y_k \right)^\kappa - \left(\sum_{k:A_{ik} < A_{ij}} y_k \right)^\kappa \right]}{\sum_{k:A_{ik}=A_{ij}} y_k} \tag{7}$$

$$w_i = \sum_j \frac{B_{ji} x_j \left[\left(\sum_{k:B_{ki} \leq B_{ji}} x_k \right)^\kappa - \left(\sum_{k:B_{ki} < B_{ji}} x_k \right)^\kappa \right]}{\sum_{k:B_{ki}=B_{ji}} x_k} \tag{8}$$

$$\frac{dx_i}{dt} = \frac{\alpha x_i}{\tau}(u_i - x^T u) + x_i \alpha \sum_j x_j ln(\frac{x_j}{x_i}) \tag{9}$$

$$\frac{dy_i}{dt} = \frac{\alpha y_i}{\tau}(w_i - y^T w) + y_i \alpha \sum_j y_j ln(\frac{y_j}{y_i}) \tag{10}$$

Since Lenient Q-learning inherits the action-value update rule from Q-learning, its behavior is similarly influenced by the asynchronous nature of the updates. In addition, the evolutionary model of Lenient Q-learning is based directly on the evolutionary model of Q-learning [8]. Therefore, the same discrepancies between predicted behavior and actual behavior are expected, and may similarly be resolved.

This article proposes the Lenient Frequency Adjusted Q-learning (LFAQ) algorithm that combines the improvements of FAQ and Lenient Q-learning. The action-value update rule of LFAQ is equal to that of FAQ; the difference is that the lenient version collects κ rewards before updating its Q-values based on the highest of those rewards. In addition to the theoretical arguments that carry over from FAQ [5], Section 4.1 provides empirical proof that LFAQ indeed matches the evolutionary model proposed by [8]. This implies that the newly proposed algorithm inherits the strong guarantees derived from the evolutionary model, i.e., the basins of attraction of the Pareto optimal equilibria can be increased to an arbitrarily large fraction of the policy space by increasing the degree of leniency.

4 Experiments

This section presents two experiments that demonstrate the validity and advantage of Lenient Frequency Adjusted Q-learning (LFAQ). First, differences between Lenient Q-learning and the evolutionary prediction are illustrated, and it is shown that the proposed LFAQ matches the evolutionary model significantly better. Second, the effect of leniency on the basins of attraction for equilibria with different payoffs is investigated. This experiment gives an intuition how leniency increases the probability of convergence to the global optimum.

4.1 Validating Lenient Frequency Adjusted Q-learning

This experiment compares the behavior of Lenient Q-learning (LQ) and Lenient FAQ-learning (LFAQ) and shows that LFAQ better matches the behavior predicted by its evolutionary model. This is done by comparing the learners' policy trajectories with the directional field of the replicator dynamics for various games. As explained in Section 3.2, the policy trajectories of LQ deviate from their expected path in a way similar to those of Q-learning. LFAQ counters this deviation by introducing an extra term in the value function update rule that compensates for the frequency with which an action is chosen.

In order to be comparable to [5], which investigates this deviation for (FA) Q-learning, a similar setup is used for the experiments. Three games are selected: the Coordination Game (CG) and the Stag Hunt (SH) game with two pure Nash equilibria; and the Matching Pennies (MP) game with one mixed Nash equilibrium. The results of the Prisoner's Dilemma which is used in [5] are not shown since MP sufficiently represents the class of competitive games, and leniency is especially suited to cooperative games. Of particular interest is the Stag Hunt game, since its two equilibria are different in nature. It has one payoff dominant

equilibrium and one risk-dominant equilibrium, and the cooperative nature of Lenient Q-learning makes it particularly suited to find this payoff dominant Nash equilibrium.

Different initializations of the Q-values result in different learning behavior in Q-learning. As LQ is a direct extension of Q-learning, a similar effect is expected. Therefore, experiments include different initial values for Q, based on the minimum (pessimistic), mean (neutral), and maximum (optimistic) possible Q-values given the game's reward space [5].

Figure 2 shows a combination of the RD directional field plots and the policy trajectories of both LQ and LFAQ for the three different games and initialization settings. Both learners use $\tau = 0.1$, $\gamma = 0.9$ and $\kappa = 5$. A learning rate of 10^{-5} is chosen in order to obtain predictable behavior, which with the given value for κ relates to $\alpha = 5 \cdot 10^{-5}$ for LQ, and $\alpha = 5 \cdot 10^{-2}$ and $\beta = 10^{-3}$ for LFAQ.

These results show that the behavior of LQ indeed depends on the initial Q-values. This leads to significant differences in convergence properties of the learner. For example, the pessimistic and neutral initialization in the Matching Pennies lead to outward movement, which actually means moving away from the equilibrium. In the Stag Hunt and Coordination Game, the learner converges to either one of the equilibria depending on the initial settings. LFAQ is more robust; it behaves the same irrespective of the initialization. In each of the cases LFAQ converges to the game's Nash equilibrium, and in the cooperative games the learning traces converge to the payoff dominant Nash equilibrium. The fact that LFAQ does not quite reach the equilibrium point in the Stag Hunt is caused by the temperature setting; decreasing the temperature allows the algorithm to reach the equilibrium.

Comparing the learners' trajectories to the evolutionary model, it is clear that LQ deviates from the expected path in most cases, whereas LFAQ shows behavior consistent with the predicted dynamics. The learners behave most similar to each other with maximum initial Q-values. However, in many applications the rewards are not known in advance, making it impossible to calculate the maximum Q-value for a game. LFAQ is a better choice than LQ since its behavior does not depend on the initialization and is consistent with the evolutionary prediction.

4.2 An example of lenient convergence to the global optimum

The main challenge to converge to a Pareto optimal solution in cooperative games is mis-coordination, which may lead to inferior rewards. As long as actions are not sufficiently coordinated, the probability of lower rewards is high and the agent may have more incentive to adopt a safe action. This can be demonstrated using the example of the Stag Hunt game, where mis-coordination on the Pareto optimum is punished, while the alternative action yields a mediocre reward that is independent of the other agent's action. Lenient learning ignores lower rewards and thereby increases the probability of converging to a Pareto optimal solution.

Figure 3 shows the dynamics of LFAQ in the Stag Hunt game given different degrees of leniency. The highest payoff is achieved by playing the first action with probability one. The basin of attraction for the global optimum grows with the degree of leniency, and in the limit consumes the whole strategy space.
These results illustrate the claim of [8] that "properly-set lenient learners are guaranteed to converge to the Pareto-optimal Nash equilibria in coordination games". In combination with the previous result, this demonstrates the value of the newly proposed Lenient Frequency Adjusted Q-learning algorithm, which inherits the theoretical guarantees from the evolutionary model.

5 Discussion and conclusions

The contributions of this article are three-fold: 1. the Lenient Frequency Adjusted Q-learning algorithm has been introduced, 2. it has been compared with Lenient Q-learning and its evolutionary model, and 3. the ability of LFAQ to converge to the Pareto optimal Nash equilibrium has been illustrated.

The proposed LFAQ algorithm combines insights from FAQ [5] and LQ [8] and inherits the advantages of both. Empirical comparisons confirm that the LFAQ algorithm is consistent with the evolutionary model derived by [8], whereas the LQ algorithm may deviate considerably. Furthermore, the behavior of LFAQ is independent of the initialization of the Q-values. This implies that it is less prone to problems arising from premature temperature decrease, which yields similar artifacts as skewed initialization, and it is overall easier to set up. In addition, the behavior of LFAQ is more desirable than the behavior of the original LQ method with respect to the learning trajectories followed. Finally, the probability of converging to a Pareto optimal equilibrium with LFAQ can be increased arbitrarily close to one by raising the degree of leniency.

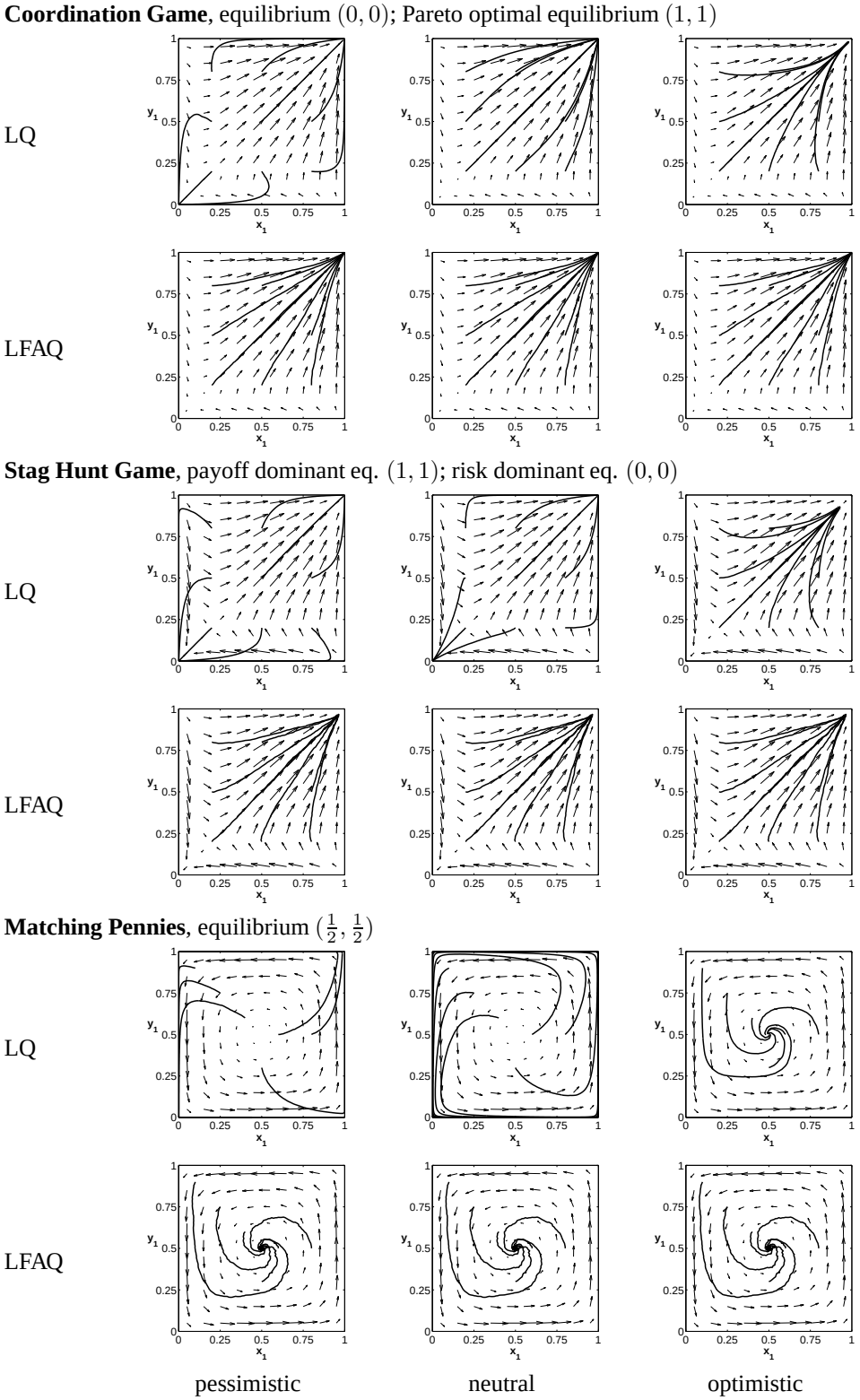

Figure 2: Overview of the behavior of Lenient Q-learning and Lenient FAQ-learning in different games. The figure shows different initialization settings for the Q-values: pessimistic (left), neutral (center) and optimistic (right). The arrows represent the directional field plot of the replicator dynamics of Lenient FAQ-learning.

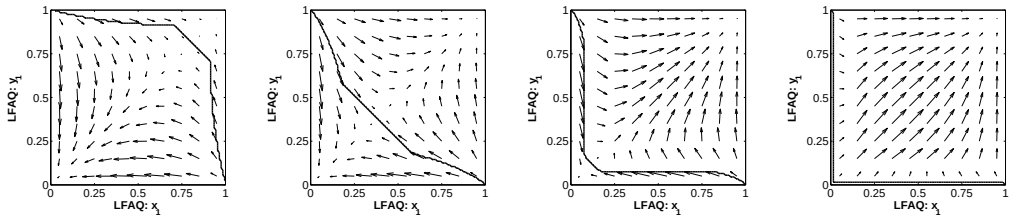

Figure 3: Lenient Frequency Adjusted Q-learning in the Stag Hunt game with varying degree of leniency. The line indicates the border between the two basins of attraction for the risk dominant equilibrium at $(0, 0)$ and the payoff dominant equilibrium at $(1, 1)$.

The methodology underlying this article demonstrates the value of evolutionary game theory as a tool to analyze, understand and design multi-agent learning algorithms. Future work will provide an empirical evaluation of the performance of lenient learners against lenient and non-lenient learners in a variety of normal form games.

References

[1] T. Börgers and R. Sarin. Learning through reinforcement and replicator dynamics. *Journal of Economic Theory*, 77:1–14, 1997.

[2] L. Busoniu, R. Babuška, and B. De Schutter. A comprehensive survey of multi-agent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 38(2):156–172, 2008.

[3] R. Gibbons. *A Primer in Game Theory*. Pearson Education, 1992.

[4] H. Gintis. *Game Theory Evolving*. University Press, Princeton, NJ, 2nd edition, 2009.

[5] M. Kaisers and K. Tuyls. Frequency adjusted multi-agent Q-learning. In van der Hoek, Kamina, Lespérance, Luck, and Sen, editors, *Proc. of 9th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 309–315, May, 10-14, 2010.

[6] J. Maynard Smith and G. R. Price. The logic of animal conflict. *Nature*, 246(2):15–18, 1973.

[7] L. Panait, K. Sullivan, and S. Luke. Lenience towards teammates helps in cooperative multiagent learning. In Nakashima, Wellman, Weiss, and Stone, editors, *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2006)*, 2006.

[8] L. Panait, K. Tuyls, and S. Luke. Theoretical advantages of lenient learners: An evolutionary game theoretic perspective. *Journal of Machine Learning Research*, 9:423–457, 2008.

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

[10] K. Tuyls, P.J. 't Hoen, and B. Vanschoenwinkel. An evolutionary dynamical analysis of multi-agent learning in iterated games. *Journal of Autonomous Agents and Multi-Agent Systems*, 12(1):115–153, 2006.

[11] K. Tuyls, K. Verbeeck, and T. Lenaerts. A selection-mutation model for q-learning in multi-agent systems. In *Proceedings of AAMAS 2003, The ACM International Conference Proceedings Series*, 2003.

[12] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.