Theoretical Advantages of Lenient Q-learners: An Evolutionary Game Theoretic Perspective

Liviu Panait
Google Inc
604 Arizona Ave, Santa Monica, CA, USA
liviu@google.com

Karl Tuyls
Maastricht University
MiCC-IKAT, The Netherlands
k.tuyls@micc.unimaas.nl

ABSTRACT

This paper presents the dynamics of multiple reinforcement learning agents from an Evolutionary Game Theoretic (EGT) perspective. We provide a Replicator Dynamics model for traditional multiagent Q-learning, and we extend these differential equations to account for lenient learners: agents that forgive possible mistakes of their teammates that resulted in lower rewards. We use this extended formal model to visualize the basins of attraction of both traditional and lenient multiagent Q-learners in two benchmark coordination problems. The results indicate that lenience provides learners with more accurate estimates for the utility of their actions, resulting in higher likelihood of convergence to the globally optimal solution. In addition, our research supports the strength of EGT as a backbone for multiagent reinforcement learning.

1. INTRODUCTION

Reinforcement Learning (RL) is an established and profound theoretical framework for learning in single-agent systems. As long as the environment an agent experiences is stationary, and the agent can experiment enough, RL guarantees convergence to the optimal strategy [10, 11, 15].

Learning in MAS however, is a more complex and cumbersome task, which does not offer the same theoretical grounding as the single agent case. Reasons for this are plentiful: the environment in which the agents operate is typically very dynamic in nature and hard to predict, other agents are operating (and learning) in this environment as well, the reinforcement an agent receives depends on the actions taken by the other agents, and not all information is observable. All these different features make it very hard to engineer learning algorithms capable of finding optimal solutions in different situations.

One of the fundamental problems of MARL is the lack of a theoretical framework as for the single agent case. Due to the lack of such a theory, we employ an evolutionary game theoretic perspective. More precisely, our work is based on a derived formal link between the replicator dynamics equations of EGT and Q-learning, as derived in [12, 13]. The authors of these publications have illustrated that there are a number of benefits to exploiting this link: one, it provides a selection-mutation perspective of the learning process, two, the model predicts desired parameters to achieve Nash equilibriums with high utility, three, the intuitions behind a specific learning algorithm can be theoretically analysed and supported by using the basins of attraction.

In this work, we show how these equations can be used as a framework to analyze a new multiagent Q-learning algorithm, and at the same time to graphically illustrate the theoretical basins of attraction to certain equilibria in the solution space. More precisely, the equations are modified to reflect the concept of lenience introduced in [8]. Lenience occurs when an agent ignores the lower utilities that it observes. The Lenient Multiagent Q-learning algorithm (LMQ) allows agents to exhibit a time-dependent level of lenience towards their teammates. It is based on the following idea: if an agent receives rewards $r_1, r_2, ..., r_k$ when choosing action a_1 at various times during the early stages of learning, the agent ignores most of these rewards and only updates the utility of a_1 based on the maximum of $r_1, r_2, ..., r_k$. The reason for this is that those rewards were obtained while the other learning agent selected (or, better said, explored) some actions $b_1, ..., b_k$, most of which it will ignore in the future due to their lower utilities. As both agents become more selective at choosing their actions, it is expected that they will each tend to primarily select a single action (the best one). At this point, each agent should update the utility of that action based on every reward it observes. This will lower the optimistic estimation of the utility for that action until it equals the mean reward obtained by the agents for that joint reward. If this mean reward becomes lower than the estimated utility of other actions, the agent will start to explore these other actions instead. We say that the agents exhibit a time-dependent level of lenience towards their teammates.

Our findings show that typical convergence problems to suboptimal solutions are mainly caused by learners having poor estimates for the quality (utility) of their actions. Lenient learners on the other hand account for more accurate estimates, and perform more efficiently in general. Our approach also clearly illustrates how one can use EGT as a tool to design a new RL algorithm, and how to analyze its strengths.

The remainder of this paper is structured as follows. In Section 2 we elaborate on the necessary background to understand further developments of this paper. More precisely, we introduce the Replicator Dynamics and Q-learning, and formally derive the link between both. In Section 3 we extend the EGT model with the concept of lenience and show how it can be computed. Section 4 addresses the experimental setup and discusses the results. Section 5 concludes the paper.

2. BACKGROUND

In this section, we describe the necessary background to understand the remainder of the paper. More precisely, we briefly introduce EGT and elaborate on its Replicator Dynamics. We continue by explaining Q-learning, a value-function based approach to reinforcement learning, in which *Q*ualities over state-action pairs are learned. We end this section with the derivation of the formal relationship between Q-learning and EGT, necessary to understand the extensions in Section 3.

2.1 Evolutionary Game Theory

Originally, Evolutionary Game Theory was proposed by John Maynard-Smith, when he relaxed the (too) strong premises behind Game Theory (GT) and applied it to biology [6, 7]. More precisely, Classical GT is a normative theory, in the sense that it expects agents to be perfectly rational and behave accordingly [9, 14, 17]. It is also the mathematical study of interactive decision making in the sense that the agents involved in the decisions take into account their own choices and those of others. Choices are determined by stable preferences concerning the outcomes of agents' possible decisions, and, also by strategic interactions involving agents that take into account the relation between their own choices and the decisions of other agents. Agents in the classical setting have a perfect knowledge of the environment and the payoff tables, and try to maximize their individual payoff. However, under these new biological circumstances, considered by Maynard-Smith, it becomes impossible to judge what choices are the most rational ones. The question now becomes how an agent can learn to optimize its behaviour and maximize its return. This learning process matches the concept of evolution in Biology as will be shown in subsequent sections.

There are two key concepts in EGT, one, evolutionary stable strategies (ESS), and two, the replicator dynamics (RD). ESS is actually a refinement of the Nash equilibrium concept from classical GT. We will not discuss it further in this paper. The RD are a system of differential equations describing how a population of different strategies evolves through time. An abstraction of an evolutionary process usually combines two basic elements: selection and mutation. Selection favors some varieties over others, while mutation provides variety in the population. The most basic form of the RD only highlights the role of selection, i.e., how are the better strategies in a population selected.

The general form of a replicator dynamic is the following:

$$\frac{dx_i}{dt} = [(A\mathbf{x})_i - \mathbf{x}\cdot A\mathbf{x}]x_i \tag{1}$$

In equation (1), x_i represents the density of strategy i in the population, and A is the payoff matrix that describes the different payoff values that each individual replicator receives when interacting with other replicators in the population. The state of the population ($\mathbf{x}$) can be described as a probability vector $\mathbf{x} = (x_1, x_2, ..., x_J)$ which expresses the different densities of all the different types of replicators in the population. Hence $(A\mathbf{x})_i$ is the payoff that replicator i receives in a population with state x and $\mathbf{x}\cdot A\mathbf{x}$ describes the average payoff in the population. The growth rate $\frac{\frac{dx_i}{dt}}{x_i}$ of the population share using strategy i equals the difference between the strategy's current payoff and the average payoff in the population. For further information we refer the reader to [2, 3, 17].

In this paper, however, we are concerned with formal models of multiple agents that learn concurrently. For simplicity, we restrict the discussion to only two such learning agents. As a result, we need two systems of differential equations, one for each agent. This setup corresponds to a RD for asymmetric games. If B is the payoff matrix that describes the payoff values received by the second agent, and if $A = B^t$, then equation (1) would emerge again to characterize the dynamics of the second learner.

This translates into the following replicator equations for the two populations:

$$\frac{dp_i}{dt} = [(A\mathbf{q})_i - \mathbf{p}\cdot A\mathbf{q}]p_i \tag{2}$$

$$\frac{dq_i}{dt} = [(B\mathbf{p})_i - \mathbf{q}\cdot B\mathbf{p}]q_i \tag{3}$$

As can be seen in equation (2) and (3), the growth rate of the types in each population is additionally determined by the composition of the other population, in contrast to the single learner case described by equation (1).

2.2 Q-learning

The learners we consider in this paper employ a specific reinforcement learning algorithm known as Q-learning [16]. Q-learning is particularly useful in multiagent domains where reinforcement information (expressed as penalties or rewards) is stochastic and is observed after a sequence of actions has been performed. Q-learning associates a utility Q with each (s,a) pair, where s is a state of the environment, and a is an action that the agent can perform when in state s. The agent updates the Q values at each time step based on the reinforcement it has received. The update formula is

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left(r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)\right) \tag{4}$$

where the agent has just been in state s_t, has performed action a_t, has observed reward r_t, and it has transitioned to the new (current) state s_{t+1}; α is the learning rate, and γ is a discount factor for incorporating future rewards into the utility estimate. Action selection is usually based on a stochastic process; popular choices include the ε-greedy exploration (select the best action with probability $1-\varepsilon$, or a random action otherwise) and the Boltzmann exploration (the selection process further depends on a "temperature" parameter). As previously mentioned, Q-learning has certain guarantees of convergence under theoretical conditions, which include performing each action an infinite number of times in each state, as well as proper settings for the learning rate [10, 15].

Multiagent Q-learning represents a straightforward extension of Q-learning to domains involving multiple agents [1]. Here, each agent is given its own Q-value function for (state,action) pairs. All agents choose their actions independently and concurrently, perform them in parallel, and all observe the same reward associated with the joint action. In contrast to single-agent Q-learning, the information an agent observes depends on the actions chosen by other agents. The reward information is then used by each agent to update its Q-values. Agents usually cannot perceive which actions their teammates have chosen.

Straightforward applications of multiagent Q-learning have revealed problematic convergence problems to suboptimal solutions (for example, in [1]). This significantly contrasts the well-understood convergence guarantees of Q-learning in single-agent domains. In this paper, we present an RD model for multiagent Q-learning, and we use it to show that the convergence to suboptimal solutions is caused by learners having poor estimates for the quality (utility) of their actions. We also extend this model to account for accurate utility estimates, and we demonstrate the benefits of these changes to the performance of the algorithm.

To simplify the theoretical analysis, we assume that agents choose actions by using the Boltzmann selection: an action a_k is chosen with probability

$$P(a_k) = \frac{e^{\frac{Q(s,a_k)}{\tau}}}{\sum_i e^{\frac{Q(s,a_i)}{\tau}}} \qquad (5)$$

where τ is a temperature parameter used to balance exploration and exploitation (the agent tends to select actions associated with higher utilities when τ is low).

2.3 A Formal model of Q-learning Dynamics

In this section we concisely present the formal relation between Q-learning and the RD from EGT. The reader who is interested in the complete derivation of the model is referred to [12, 13]. Basically, the derivation boils down to constructing a continuous time limit of the Q-learning model, where Q-values are interpreted as Boltzmann probabilities for the action selection. For simplicity, we only consider games between two agents, and we assume that the game is stateless: the reward that the agents receive depends solely on the actions they have currently performed in the environment. We admit upfront that such scenarios are unrealistic for practical purposes, but they are complex enough to emphasize specific challenges for multiagent reinforcement learning algorithms. For this reason, several previous empirical investigations of these techniques have employed such domains (for example, in [1, 4, 5]).

Each agent has a probability vector over its action set, more precisely $x_1,...,x_n$ over action set $a_1,...,a_n$ for the first agent and $y_1,...,y_m$ over $b_1,...,b_m$ for the second agent. We rewrite Equation 5 as follows:

$$x_i(k) = \frac{e^{\frac{Q_{a_i}(k)}{\tau}}}{\sum_{j=1}^n e^{\frac{Q_{a_j}(k)}{\tau}}}$$

where $x_i(k)$ is the probability of playing strategy i at time step k and τ is the temperature.
Now we can find an expression for, $x_i(k+1)$.

$$\begin{aligned}\frac{x_i(k+1)}{x_i(k)} &= \frac{e^{\frac{Q_{a_i}(k+1)}{\tau}} \sum_j e^{\frac{Q_{a_j}(k)}{\tau}}}{e^{\frac{Q_{a_i}(k)}{\tau}} \sum_j e^{\frac{Q_{a_j}(k+1)}{\tau}}} \\ &= \frac{e^{\frac{\Delta Q_{a_i}(k)}{\tau}}}{\sum_j x_j(k) e^{\frac{\Delta Q_{a_j}(k)}{\tau}}}\end{aligned}$$

From which follows,

$$x_i(k+1) = x_i(k) \frac{e^{\frac{\Delta Q_{a_i}(k)}{\tau}}}{\sum_j x_j(k) e^{\frac{\Delta Q_{a_j}(k)}{\tau}}}$$

If we consider the difference equation for x_i we have,

$$\begin{aligned}x_i(k+1) - x_i(k) &= \frac{x_i(k) e^{\frac{\Delta Q_{a_i}(k)}{\tau}}}{\sum_j x_j(k) e^{\frac{\Delta Q_{a_j}(k)}{\tau}}} - x_i(k) \\ &= x_i(k)\left(\frac{e^{\frac{\Delta Q_{a_i}(k)}{\tau}} - \sum_j x_j(k) e^{\frac{\Delta Q_{a_j}(k)}{\tau}}}{\sum_j x_j(k) e^{\frac{\Delta Q_{a_j}(k)}{\tau}}}\right)\end{aligned}$$

For the continuous time version we suppose that the amount of time that passes between two repetitions of the game is given by δ with $0 < \delta \leq 1$. The variable $x_i(k\delta)$ describes the x-values at time $k\delta = t$. Under these assumptions, we have,

$$\begin{aligned}\frac{x_i(k\delta+\delta) - x_i(k\delta)}{\delta} &= \frac{x_i(k\delta)}{\delta \sum_j x_j(k\delta) e^{\frac{\Delta Q_{a_j}(k\delta)}{\tau}}} \times \\ &\quad (e^{\frac{\Delta Q_{a_i}(k\delta)}{\tau}} - \sum_j x_j(k\delta) e^{\frac{\Delta Q_{a_j}(k\delta)}{\tau}})\end{aligned}$$

We are interested in the limit with $\delta \to 0$. We find the state of the limit process at some time $t \geq 0$ (which we keep fixed) by taking the limit of $x_i(k\delta)$ with $\delta \to 0$ and $k\delta \to t$.

$$\begin{aligned}\lim_{\delta\to 0} \frac{\Delta x_i(k\delta)}{\delta} &= \lim_{\delta\to 0} \frac{x_i(k\delta)}{\sum_j x_j(k\delta) e^{\frac{\Delta Q_{a_j}(k\delta)}{\tau}}} \times \\ &\quad \lim_{\delta\to 0}\left(\frac{e^{\frac{\Delta Q_{a_i}(k\delta)}{\tau}}}{\delta} - \frac{\sum_j x_j(k\delta) e^{\frac{\Delta Q_{a_j}(k\delta)}{\tau}}}{\delta}\right)\end{aligned}$$

The first limit,

$$\lim_{\delta\to 0} \frac{x_i(k\delta)}{\sum_j x_j(k\delta) e^{\frac{\Delta Q_{a_j}(k\delta)}{\tau}}}$$

equals $x_i(t)$ or short x_i, because the exponent term becomes 1 ($\Delta Q_{a_j}(k\delta)$ becomes zero) and $\sum_j x_j(k\delta)$ equals 1(sum of all probabilities equals 1). Therefore,

$$\lim_{\delta\to 0} \frac{\Delta x_i(k\delta)}{\delta} = x_i \times \underbrace{\lim_{\delta\to 0}\left(\frac{e^{\frac{\Delta Q_{a_i}(k\delta)}{\tau}}}{\delta} - \frac{\sum_j x_j(k\delta) e^{\frac{\Delta Q_{a_j}(k\delta)}{\tau}}}{\delta}\right)}_{T_2}$$

The second limit is undefined (we have a $\frac{0}{0}$ situation), which allows us to use the rule of the l'hôspital. The second term now equals (for short T_2),

$$T_2 = \frac{dQ_{a_i}(t)}{\tau dt} - \sum_j x_j(t) \frac{dQ_{a_j}(t)}{\tau dt}$$

The total limit now becomes,

$$\frac{\frac{dx_i}{dt}}{x_i} = \frac{1}{\tau}\left(\frac{dQ_{a_i}}{dt} - \sum_j \frac{dQ_{a_j}}{dt} x_j\right) \qquad (6)$$

We now have derived the continuous time model of the Q-learning process. As you can see in equation (6) we need an expression for $\frac{dQ_{a_i}(t)}{dt}$. We can derive the differential equation for the Q-function by performing the following steps.

The Q-learning update rule for the first agent can be written as follows,

$$Q_{a_i}(k+1) = Q_{a_i}(k) + \alpha(r_{a_i}(k+1) + \gamma \max_{a_i} Q - Q_{a_i}(k))$$

which implies,

$$\Delta Q_{a_i}(k) = \alpha(r_{a_i}(k+1) + \gamma \max_{a_i} Q - Q_{a_i}(k))$$

This expression is the difference equation for the Q-function. If we make this equation infinitesimal, going from discrete steps to a continuous version, we suppose that the amount of time that passes between two repetitions of updates of the Q-values is given by δ with $0 < \delta \leq 1$. The variable $Q_{a_i}(k\delta)$ describes the Q-values at time $k\delta$. Now, we get,

$$\Delta Q_{a_i}(k\delta) = \alpha(r_{a_i}((k+1)\delta) + \gamma \max_{a_i} Q - Q_{a_i}(k\delta))((k+1)\delta - k\delta)$$

which is the same as writing,

$$\Delta Q_{a_i}(k\delta) = \alpha(r_{a_i}((k+1)\delta) + \gamma \max_{a_i} Q - Q_{a_i}(k\delta))\delta$$

We are interested in the limit $\delta \to 0$. We find the state of the limit process at some time $t \geq 0$ (which we keep fixed) by taking the limit of $Q_{a_i}(k\delta)$ with $\delta \to 0$ and $k\delta \to t$. If we now bring δ to the left side, and take the limit for $\delta \to 0$, we have

$$\frac{dQ_{a_i}}{dt} = \alpha(r_{a_i} + \gamma \max_{a_i} Q - Q_{a_i}) \tag{7}$$

Now, we can substitute equation (7) in equation (6), yielding,

$$\frac{\frac{dx_i}{dt}}{x_i} = \frac{1}{\tau}\alpha(r_{a_i} - \sum_j x_j r_{a_j} - Q_{a_i} + \sum_j Q_{a_j} x_j)$$

because $\sum_j x_j = 1$, this expression becomes,

$$\frac{\frac{dx_i}{dt}}{x_i} = \frac{\alpha}{\tau}(r_{a_i} - \sum_j x_j r_{a_j} + \sum_j x_j (Q_{a_j} - Q_{a_i}))$$

Given that $\frac{x_j}{x_i}$ equals $\frac{e^{\frac{Q_{a_j}}{\tau}}}{e^{\frac{Q_{a_i}}{\tau}}}$, it follows that

$$\alpha \sum_j x_j ln(\frac{x_j}{x_i}) = \frac{\alpha}{\tau}\sum_j x_j (Q_{a_j} - Q_{a_i})$$

which gives us,

$$\frac{\frac{dx_i}{dt}}{x_i} = \frac{\alpha}{\tau}(r_{a_i} - \sum_j x_j r_{a_j}) + \alpha \sum_j x_j ln(\frac{x_j}{x_i})$$

Now let us put this equation in the context of the two concurrent learning agents. For clarity, we use u_i instead of r_{a_i} to indicate the expected reward that the first agent expects for performing action i, and we use w_j to indicate the expected reward that the second agent expects for performing action j. Remember that the payoff matrix for the second agent is simply the transposed payoff matrix used by the first agent.

We have that $u_i = \sum_k a_{ik} y_k$ is the expected reward that would be observed by the first agent, given the other agent's current probabilities of selecting among its actions (and similarly $w_j = \sum_k a_{kj} x_k$). The RD model above has therefore the simpler form,

$$u_i = \sum_k a_{ik} y_k \tag{8}$$

$$w_j = \sum_k a_{kj} x_k \tag{9}$$

$$\frac{\frac{dx_i}{dt}}{x_i} = \frac{\alpha}{\tau}\left(u_i - \sum_k x_k u_k\right) + \alpha \sum_k x_k \ln \frac{x_k}{x_i} \tag{10}$$

$$\frac{\frac{dy_j}{dt}}{y_j} = \frac{\alpha}{\tau}\left(w_j - \sum_k y_k w_k\right) + \alpha \sum_k y_k \ln \frac{y_k}{y_j} \tag{11}$$

3. FORMAL MODELS OF LENIENT Q-LEARNERS

The RD model described in Section 2.3 updates the utility of an action based on the expected reward for that action in combination with all possible actions that the teammate might select. This is a conservative setting, and as a result the agent might be biased toward selecting actions that receive good rewards on average. We argue that this may not be the best approach in a cooperative scenario. Consider the following example involving two agents A and B that learn to play soccer together, and suppose that A has possession of the ball. One action b_1 that B might perform in this case is to get itself unmarked (such that it can receive the ball from A) and in an advantageous situation (maybe closer to the opponents' goal, from where it could score). A second action b_2 is to adopt a defensive position[1] to defend in case A decides to attempt to score in its own goal, or if A passes the ball to the opponent and does little to defend afterwards. A significant problem that learner B might face is that A could do a lot of these mistakes, especially at early stages of learning; and a consequence of these mistakes might be that B observes that the expected reward that it receives upon performing b_1 is lower than the one it receives for b_2. The result of this estimation of expected reward is that agents might tend to prefer very conservative actions that work well with a large variety of actions for the teammate, but which may also result in suboptimal (and usually undesirable) joint actions for the team of agents.

One solution to remedy this problem is to allow agents to show lenience towards their teammates: the agents can *ignore* lower rewards observed upon performing their actions, and only update the utilities of actions based on the higher rewards. This can be simply achieved if the learners compare the observed reward with the estimated utility for an action, and update the utility only if it is lower than the reward. In some sense, such an approach changes the learning focus from performing as well as possible in the context of the current (most likely mediocre) teammate, to performing as well as possible in the context of improved behaviors for its teammate (as the teammate is expected to have as learning progresses).

The advantages of such an approach were demonstrated empirically in [8]. The authors propose a time-dependent degree of lenience: the agents start by ignoring more of the lower reward that they observe, but as learning progresses, agents tend to explore certain "better" actions and also to ignore fewer lower rewards. Panait et al. present empirical evidence that several multiagent learning paradigms can significantly benefit from agents being lenient to one another.

Here, we concentrate on the mathematical foundations of a simpler approach to lenient learners: each agent collects the rewards it receives for performing actions. Upon observing N rewards for an action, only the maximum of these N rewards is used to update the utility associated with that action. The set of observed rewards for that action is cleared, and the learning process continues. In other words, the agents employ a fixed degree of lenience (which is determined by N) to their teammates. The more lower rewards the agents ignore (the higher N is), the more lenient the agents can be said to be. Although this approach does not model Panait et al's algorithms, we believe its formal analysis has two major contributions. First, it provides a valuable addition to our set of tools to study such multiagent learning algorithms, and also strengthens Evolutionary Game Theory as a primary framework for such analysis. Second, it adds to our understanding of what are the causes for multiagent learning algorithms drifting away from globally optimal solutions.

Next, we extend the RD model from Section 2.3 such that each learner uses only the maximum of N rewards (it ignores the lower $N-1$ rewards) to improve its utility estimate. The following theorem establishes a key result for the formal model of lenient Q-learners.

[1]A more extreme variation could be for B to proactively attempt to take possession of the ball away from its teammate, in order to prevent it from making a possible mistake.

THEOREM 1. *Let a_{ij} be the payoff for joint action (i,j), $(p_j)_{j\in 1..n}$ be the probability that the teammate selects action j. The expected maximum payoff for i over N pairwise combinations with actions $j_1 \dots j_N$ chosen with replacement according to $(p_j)_{j\in 1..n}$ is* $\sum_{j=1}^{n} a_{ij} \frac{p_j}{\sum_{k:a_{ik}=a_{ij}} p_k} \left(\left(\sum_{k:a_{ik}\le a_{ij}} p_k\right)^N - \left(\sum_{k:a_{ik}<a_{ij}} p_k\right)^N \right)$.

PROOF. The expected maximum reward of i over N pairwise combinations can be expressed as a weighted sum of all possible rewards a_{ij} that i can receive with different actions j for the teammate. The weight of each term a_{ij} equals the probability that a_{ij} is the maximum reward observed over N trials. This probability can be computed based on the difference between the probability of receiving N rewards that are no higher than a_{ij}, which equals $\left(\sum_{k:a_{ik}\le a_{ij}} p_k\right)^N$, minus the probability of receiving N rewards that are strictly lower than a_{ij}, which equals $\left(\sum_{k:a_{ik}<a_{ij}} p_k\right)^N$. Observe that an action i might receive the same reward in combination with different actions of the teammate, and as a result, there is an extra weight $\frac{p_j}{\sum_{k:a_{ik}=a_{ij}} p_k}$ that computes the probability of observing the reward a_{ij} in combination with action j of the teammate, out of all the other actions the teammate might have selected and would have resulted in the same reward. □

We infer an RD model for lenient Q-learners by combining the RD model for traditional Q-learners in equations (8) – (11) with the result in Theorem 1. The result is a set of four equations:

$$u_i = \sum_{j=1}^{m} \frac{a_{ij} y_j \left(\left(\sum_{k:a_{ik}\le a_{ij}} y_k\right)^N - \left(\sum_{k:a_{ik}<a_{ij}} y_k\right)^N \right)}{\sum_{k:a_{ik}=a_{ij}} y_k} \quad (12)$$

$$w_j = \sum_{i=1}^{n} \frac{a_{ij} x_i \left(\left(\sum_{k:a_{kj}\le a_{ij}} x_k\right)^N - \left(\sum_{k:a_{kj}<a_{ij}} x_k\right)^N \right)}{\sum_{k:a_{kj}=a_{ij}} x_k} \quad (13)$$

$$\frac{\frac{dx_i}{dt}}{x_i} = \frac{\alpha}{\tau}\left(u_i - \sum_k x_k u_k\right) + \alpha \sum_k x_k \ln \frac{x_k}{x_i} \quad (14)$$

$$\frac{\frac{dy_j}{dt}}{y_j} = \frac{\alpha}{\tau}\left(w_j - \sum_k y_k w_k\right) + \alpha \sum_k y_k \ln \frac{y_k}{y_j} \quad (15)$$

Note that the two equations describing the update rule for the two learners have not changed. What has changed, however, is the expected reward that is used to update the utilities of actions (equations (12) and (13)). As expected, observe that the two RD models described by equations (8) – (11) and (12) – (15) are equivalent for $N = 1$ (when the agents do not ignore any lower reward). For this reason, we use the setting $N = 1$ as a benchmark representing the formal model for traditional Q-learners.

4. ANALYSIS: BASINS OF ATTRACTION

This section demonstrates the advantages of improving the learners' estimates for the quality of their actions. To this end, we visualize the basins of attraction to suboptimal and optimal Nash equilibria for the extended RD model of lenient Q-learners. As noted before, our experiments also involve traditional multiagent Q-learning approaches for the $N = 1$ setting.

We use the formal RD model to study the behavior of lenient learners in two multiagent coordination games: Climb and Penalty. Climb has two Nash equilibria, the optimum $(1,1)$ and the suboptimum $(2,2)$, and it is strongly deceptive. Penalty has three Nash equilibria: $(1,1)$, $(2,2)$, and $(3,3)$. Of them, $(2,2)$ is suboptimal, but is more forgiving if one or the other population deviates from the Nash equilibrium. The problem domains are similar to the ones introduced in [1] and used in previous investigations of multiagent Q-learning [4, 5]. The joint payoff matrices for these coordination games are defined as:

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

Observe that these simple domains preserve challenges associated with multiagent learning, as in the example at the beginning of Section 3. Consider the simple situation where agents are equally likely to choose either of their three actions in the Penalty domain. A traditional Q-learner will estimate a zero expected reward for actions 1 and 3, while the expected reward for action 2 is 0.66. Thus, traditional Q-learners might often have higher utility estimates for actions corresponding to suboptimal solutions, resulting in an unfortunate convergence to inferior Nash equilibria.

The visualization works as follows. First, we project the space of initial configurations of the multiagent Q-learning system (the probabilities for agents' selecting each of their actions) along the edges of a square: the vertical edge represents a projection of the initial configurations for the first learner, and the initial configurations for the second learner are projected on the horizonal axis. We iterate the RD model starting from each pair of initial configurations for the two agents. Observe that the RD model is completely deterministic and it only depends on the current state of the multiagent system. As a consequence, the RD model is expected to converge upon iteration to a Nash equilibrium in the payoff matrix. We use different colors to encode to which equilibrium the system has converged: black dots represent initial configurations that resulted in convergence to the suboptimal Nash equilibrium, while white and grey dots represent initial configurations that made the system converge to the global optimum. Of course, better multiagent Q-learning system will have fewer black dots.

Observe that the space of initial configurations for each learner is the simplex $\Delta^3 = \left\{(p_1,p_2,p_3) \in [0,1]^3 \mid \sum_{i=1}^{3} p_i = 1\right\}$. The projection of Δ^3 to one dimension starts by dividing it into six equal-area triangles, as in Figure 1 (left). Initial configurations in areas $1-2$ have $p_1 \ge \max(p_2,p_3)$, and similarly areas $3-4$ and $5-6$ have $p_2 \ge \max(p_1,p_3)$ and $p_3 \ge \max(p_1,p_2)$, respectively. The areas are projected in increasing order (all initial configurations in area 1 are projected first, and so on). Inside each area, initial configurations are ordered lexicographically in the direction of the arrow. More specifically, in regions $1-2$, the sorting is done primarily on p_1, and secondarily on p_2; for $3-4$, p_2 and p_3; for $5-6$, p_3 and p_1. Even-numbered regions are sorted ascending and odd-numbered regions are sorted descending. The objective of all these procedures is to group together regions of the space of initial configurations that are expected to converge to the same Nash equilibrium. We sample 216 initial configurations in the simplex: the six areas in Figure 1(left) are each divided into six triangles, and each of them is further divided into six more triangles. The center of each resulting triangle corresponds to an initial configuration. We add random noise distributed uniformly between -0.00005 and 0.00005 to reduce certain artifacts due to identical settings for the two agents. The sampling also does not cover initial configurations on the edges or vertexes of the simplex, but the probability that an evolutionary algorithm starts from those initial configurations is 0 anyway.

The right image in Figure 1 is an example of the resulting pro-

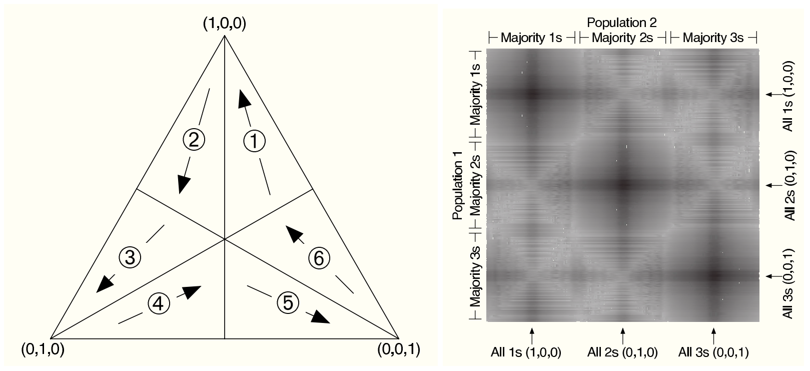

Figure 1: The projection of $\Delta^3 \times \Delta^3$ to $[0,1]^2$. (left): The projection divides the simplex Δ^3 into six equal-area triangles; arrows shows the direction for sorting points in each area. (right): Visualization of the cartesian product of two simplexes.

jection of $(\Delta^3)^2$ onto 2-D. Thanks to the sorting described above, certain regions reflect majority-1, majority-2, and majority-3 regions; and borders between those regions are the mixture of the two regions. Dark lines in the figure show locations that have high ratios of 1s, 2s, or 3s in one or the other configuration.

The rectangle in Figure 1(right) contains $216^2 = 46656$ dots: each visualization of the basins of attraction will be computed based on 46656 runs of the RD model. Given that the RD model provides differential equations, we approximate the next state of the concurrent learning system by assuming the variations in the derivative are small over short periods of time θ:

$$\begin{aligned} x_i' &= x_i + \theta \frac{dx_i}{dt} \\ y_j' &= y_j + \theta \frac{dy_j}{dt} \end{aligned}$$

We set $\theta = 0.001$, the learning rate $\alpha = 0.1$, and the exploration parameter $\tau = 0.01$. We iterate the RD model 100000 times, or until both agents have a probability exceeding $1 - 10^{-10}$ to select one of their actions over the other two.

The basins of attraction for the optimal $(1,1)$ (both play their first strategy) and suboptimal $(2,2)$ (both play their second strategy) equilibria in the Climb domain are visualized in Figure 2. Given that the new RD model reduces to the one described in [12] when $N = 1$, the basin of attraction in the top-left graph indicates that a lot of trajectories for straightforward extensions of Q-learning to multiagent domains will often converge to the suboptimal solution. As the learners ignore more of the lower rewards, they improve their estimates for the quality of their actions, and are thus more likely to converge to the global optimum. The same behavior can be observed in the Penalty domain, as illustrated in Figure 3.

We visualize the impact of improved utility estimates due to increasing levels of lenience for each learner. Figure 2 shows the basins of attraction in the Climb coordination game. The images show that the difficulty of the problem domain decreases as each population is provided with more accurate estimates for the utilities of actions. When updating utilities based on all rewards (the original RD model), it appears that the multiagent Q-learning search will find the optima if at least one of the agents starts with a high probability of selecting action 1. Even in this case, the system is most likely to converge to the global optimum if the probability of selecting action 2s is relatively low. As each agent gets a better estimate of action utility (via using the maximum of more observed rewards), the basin of attraction for the suboptimal equilibria reduces to areas where at least one of the initial agents has a high probability of selecting actions 2 or 3: the higher N (the lenient the agents are), the larger the initial probability required to still converge to the sub-optimum.

Figure 3 presents the basins of attraction in the Penalty game. Observe that the two global optima cover most of the space even when a single collaborator is used; the suboptimal equilibria covers mainly areas where at least one of the agents starts with a high probability of selecting action 2, and the other agent has equal probabilities of selecting actions 1 and 3 — this increases the percentage of miscoordinations. As N increases, the basin of attraction for the suboptimal Nash equilibrium reduces to only areas where *both* agents start with extremely high initial probabilities of selecting action 2. The visualization of the basins of attraction suggests that Penalty is a much easier coordination game than Climb. Note also a thin diagonal line in the top-left graph of Figure 3. Interestingly, this is due to the fact that if the probability of one agent selecting action 1 is about equal to the probability of the other agent selecting action 3, there are frequent miscoordinations that impact on the expected reward for these actions as estimated by the learners, and the system converges to the suboptimal Nash equilibrium.

5. CONCLUSIONS AND FUTURE WORK

This paper presented an extended formal model for a new class of multiagent learning algorithms, namely those involving lenient agents that ignore low rewards that they observe upon performing actions in the environment. We also detailed a visualization technique to illustrate the basins of attraction to different optimal and suboptimal Nash equilibria, and exemplified how intuitive graphs might reveal valuable information about the properties of multiagent learning algorithms that was lacking in the summarizations of empirical results. The paper provided theoretical support for previous reports that lenience helps learners achieve higher rates of convergence to optimal solutions, and also strengthened the use of Evolutionary Game Theory to study the properties of multiagent reinforcement learning algorithms.

There are nonetheless many avenues to continue the research presented in this paper. We plan to extend our models to help analyze the properties of multiagent learning algorithms in stochastic environments, as well as in more realistic scenarios involving states

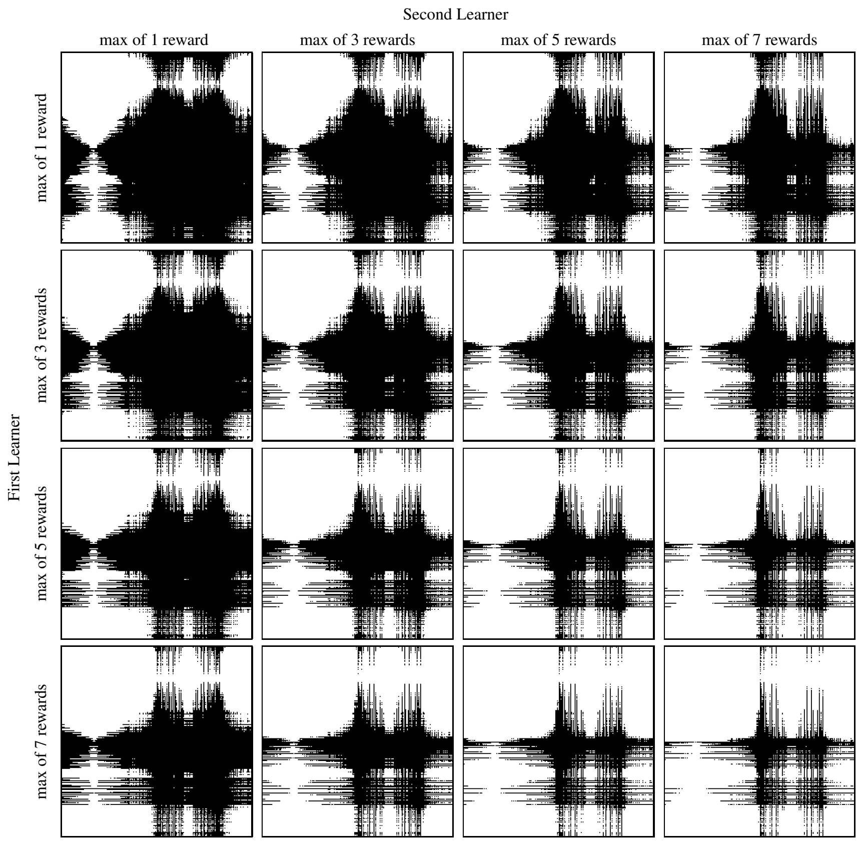

Figure 2: Basins of attraction in the Climb problem domain for multiagent Q-learning that updates the utility of an action based on the maximum of 1, 3, 5 and 7 of the rewards received when selecting that action. White and black mark the basins of attraction for the (1,1) and (2,2) equilibria.

and partial observability. We also aim to combine these models to the related EGT models for cooperative coevolutionary algorithms (see [18]), in order to analyze the similarities and differences that these multiagent learning algorithms share in common. We hope to use these models to further explore theoretical and practical solutions to multiagent learning.

6. REFERENCES

[1] C. Claus and G. Boutilier. The dynamics of reinforcement learning in cooperative multi-agent systems. In *Proceedings of the 15th International Conference on Artificial Intelligence*, pages 746–752, 1998.

[2] H. Gintis. *Game Theory Evolving: A Problem-Centered Introduction to Modeling Strategic Interaction*. Princeton University Press, 2001.

[3] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.

[4] S. Kapetanakis and D. Kudenko. Reinforcement learning of coordination in cooperative multi-agent systems. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-02)*, 2002.

[5] M. Lauer and M. Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 535–542. Morgan Kaufmann, 2000.

[6] J. Maynard-Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.

[7] J. Maynard-Smith and J. Price. The logic of animal conflict. *Nature*, 146:15–18, 1973.

[8] L. Panait, K. Sullivan, and S. Luke. Lenience towards teammates helps in cooperative multiagent learning. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi Agent Systems – AAMAS-2006*. ACM, 2006.

[9] L. Samuelson. *Evolutionary Games and Equilibrium Selection*. MIT Press, Cambridge, MA, 1997.

[10] S. P. Singh, M. J. Kearns, and Y. Mansour. Nash convergence of gradient dynamics in general-sum games. In *UAI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 541–548, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[11] R. Sutton and A. Barto. *Reinforcement Learning: An introduction.* Cambridge, MA: MIT Press ., 1998.

[12] K. Tuyls, P. 't Hoen, and B. Vanschoenwinkel. An

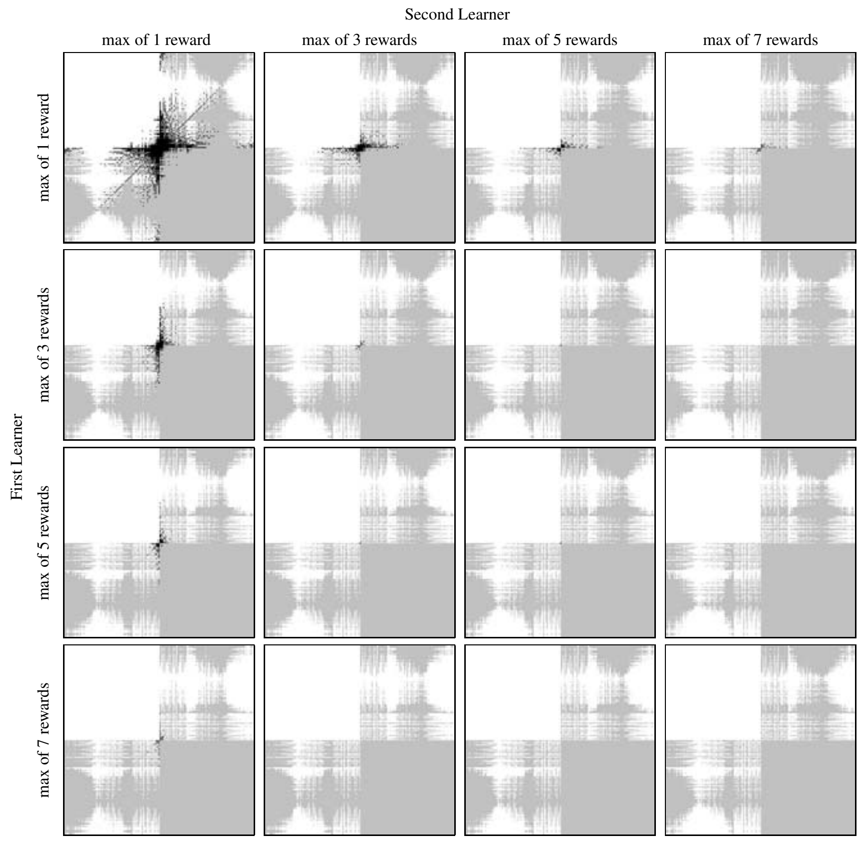

Figure 3: Basins of attraction in the Penalty problem domain for multiagent Q-learning that updates the utility of an action based on the maximum of 1, 3, 5 and 7 of the rewards received when selecting that action. White, black, and light grey mark the basins of attraction for the (1,1), (2,2), and (3,3) equilibria, respectively.

evolutionary dynamical analysis of multi-agent learning in iterated games. *The Journal of Autonomous Agents and Multi-Agent Systems*, 12:115–153, 2006.

[13] K. Tuyls, K. Verbeeck, and T. Lenaerts. A Selection-Mutation model for Q-learning in Multi-Agent Systems. In *The second International Joint Conference on Autonomous Agents and Multi-Agent Systems. ACM Press, Melbourne, Australia*, 2003.

[14] F. Vega-Redondo. *Economics and the Theory of Games.* Cambridge University Press, 2003.

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

[16] C. J. Watkins. *Models of Delayed Reinforcement Learning*. PhD thesis, Psychology Department, Cambridge University, Cambridge, United Kingdom, 1989.

[17] J. W. Weibull. *Evolutionary Game Theory.* MIT Press, 1996.

[18] R. P. Wiegand. *An Analysis of Cooperative Coevolutionary Algorithms*. PhD thesis, George Mason University, Fairfax, Virginia, 2004.