RESQ-learning in stochastic games

Daniel Hennes, Michael Kaisers and Karl Tuyls
Maastricht University
Department of Knowledge Engineering
P.O. Box 616, 6200 MD Maastricht, The Netherlands
{daniel.hennes, michael.kaisers, k.tuyls} @maastrichtuniversity.nl

ABSTRACT

This paper introduces a new multi-agent learning algorithm for stochastic games based on replicator dynamics from evolutionary game theory. We identify and transfer desired convergence behavior of these dynamical systems by leveraging the link between evolutionary game theory and multi-agent reinforcement learning. More precisely, the algorithm (RESQ-learning) presented here is the result of Reverse Engineering State-coupled replicator dynamics injected with the Q-learning Boltzmann mutation scheme. The contributions of this paper are twofold. One, we demonstrate the importance of a mathematical multi-agent learning framework by transferring insights from evolutionary game theory to reinforcement learning. Two, the resulting learning algorithm successfully inherits the convergence behavior of the reverse engineered dynamical system. Results show that RESQ-learning provides convergence to pure as well as mixed Nash equilibria in a selection of stateless and stochastic multi-agent games.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*

Keywords

Reinforcement learning, Multi-agent learning, Evolutionary game theory, Replicator dynamics, Stochastic games

1. INTRODUCTION

Modern society is characterized by a high level of interconnectedness, with the internet and mobile phone networks being the most prominent example media. As a result, most situations yield more than one actor, and should naturally be modeled as multi-agent systems to account for their inherent structure and complexity. Example applications for which significant progress has been facilitated using multi-agent learning range from auctions and swarm robotics to predicting political decisions [2, 7, 11, 13].

The learning performance of contemporary reinforcement learning techniques has been studied in great depth experimentally as well as formally for a diversity of single agent control tasks [15]. Markov decision processes provide a mathematical framework to study single agent learning. However, in general they are not applicable to multi-agent learning. Once multiple adaptive agents simultaneously interact with each other and the environment, the process becomes highly dynamic and non-deterministic, thus violating the Markov property. Evidently, there is a strong need for an adequate theoretical framework modeling multi-agent learning. Recently, a link between the learning dynamics of reinforcement learning algorithms and evolutionary game theory has been established, providing useful insights into the learning dynamics [1, 3, 17, 18]. In particular, in [1] the authors have derived a formal relation between multi-agent reinforcement learning and the replicator dynamics. This relation between replicators and reinforcement learning has been extended to different algorithms such as learning automata and Q-learning in [9, 18].

Exploiting the link between reinforcement learning and evolutionary game theory is beneficial for a number of reasons. The majority of state of the art reinforcement learning algorithms are blackbox models. This makes it difficult to gain detailed insight into the learning process and parameter tuning becomes a cumbersome task. Analyzing the learning dynamics helps to determine parameter configurations prior to actual employment in the task domain. Furthermore, the possibility to formally analyze multi-agent learning helps to derive and compare new algorithms, which has been successfully demonstrated for lenient Q-learning in [12].

However, the evolutionary game theoretic framework has been limited to either non-explorative learning in multiple states [6], or explorative single-state learning [17, 18]. The investigation of single-state learning in the latter source has shown, that exploration facilitates convergence to mixed equilibria and allows to overcome local optima, while non-explorative learning may end up in limit cycles. Therefore, this article designs a state-coupled system with the desired convergence behavior, using insights about Q-learning with Boltzmann exploration. Subsequently, this system will be reverse engineered, resulting in the derivation of *Reverse Engineered State-coupled Q-exploration* (RESQ) learning, a new multi-agent learning algorithm for stochastic games. RESQ learning is based on model-free learners with a minimum of required information (current state and reward feedback); agents maintain a policy only over their own action space. Thereby it constitutes a substantial advantage over joint-action learning approaches, such as Nash-Q [8] or Friend-or-foe (FFQ) [10]. Experiments confirm the match of the introduced algorithm with its evolutionary dynamical system. Furthermore, convergence to stable points in a selection of two-state matrix games is shown.

This paper is divided into two main parts: the forward and the reverse approach. First, Section 2 presents the forward approach, modeling multi-agent reinforcement learning

within an evolutionary game theoretic framework. Second, the inverse approach, reverse engineering the RESQ-learning algorithm is demonstrated in Section 3. Section 4 delivers a comparative study of the newly devised algorithm and its dynamics. Section 5 concludes this article.

2. FORWARD APPROACH

An adequate theoretical framework modeling multi-agent learning dynamics has long been lacking [14, 16]. Recently, an evolutionary game theoretic approach using replicator dynamics is employed to fill this gap. Replicator dynamics are a methodology of evolutionary game theory to model the dynamical evolution of strategies. Exploiting the link between reinforcement learning and evolutionary game theory is beneficial for a variety of reasons. Analyzing the learning dynamics helps to gain further insight into the learning dynamics and to determine parameter configurations before learners are actually employed in the task domain. We call this the *forward approach*.

2.1 Stateless learning dynamics

First, we focus on model free, stateless and independent learners. This means interacting agents do not model each other; they only act upon the experience collected by experimenting with the environment. Furthermore, no environmental state is considered which means that the perception of the environment is limited to the reinforcement signal. While these restrictions are not negligible they allow for simple algorithms that can be treated analytically.

2.1.1 Learning automata

A learning automaton (LA) uses the basic policy iteration reinforcement learning scheme. An initial random policy is used to explore the environment; by monitoring the reinforcement signal, the policy is updated in order to learn the optimal policy and maximize the expected reward.

The class of *finite action-set learning automata* considers only automata that optimize their policies over a finite action-set $A = \{1, \dots, k\}$ with k some finite integer. One optimization step, called *epoch*, is divided into two parts: action selection and policy update. At the beginning of an epoch t, the automaton draws a random action $a(t)$ according to the probability vector $\pi(t)$, called policy. Based on the action $a(t)$, the environment responds with a reinforcement signal $r(t)$, called reward. Hereafter, the automaton uses the reward $r(t)$ to update $\pi(t)$ to the new policy $\pi(t+1)$. The learning automaton update rule using the *linear reward-inaction* scheme is given below.

$$\pi_i(t+1) \leftarrow \pi_i(t) + \begin{cases} \alpha r\,(t)\,(1-\pi_i(t)) & \text{if } a\,(t) = i \\ -\alpha r\,(t)\,\pi_i(t) & \text{otherwise} \end{cases} \quad (1)$$

where $r(t) \in [0,1]$. The reward parameter $\alpha \in [0,1]$ determines the learning rate of the automaton.

2.1.2 Q-learning with Boltzmann exploration

In contrast to learning automata, Q-learners maintain a value estimation $Q_i(t)$ of the expected (discounted) reward for each action and are hence known as value iterators. We use Frequency Adjusted Q-learning (FAQ), a slight variation of the original Q-learning update rule [9]. The FAQ update rule with learning rate α and discount factor γ is given below.

$$Q_i(t+1) \leftarrow Q_i(t) + \min\left(\frac{\beta}{x_i}, 1\right) \cdot \alpha\left(r_i(t) + \gamma \operatorname*{argmax}_j Q_j(t) - Q_i(t)\right)$$

Again π_i denotes the probability of selecting action i. This policy is generated using a function $\pi(Q) = (\pi_1, \dots, \pi_k)$. The most prominent examples of such policy generators are ϵ-greedy and Boltzmann exploration schemes [15]. For the dynamics of ϵ-greedy Q-learning we refer to [4]. This article exclusively discusses Q-learning with Boltzmann exploration. It is defined by the following function, mapping Q-values to policies, while balancing exploration and exploitation using a temperature parameter τ:

$$\pi_i(Q,\tau) = \frac{e^{\tau^{-1}Q_i}}{\sum_j e^{\tau^{-1}Q_j}}$$

The parameter τ lends its interpretation as temperature from the domain of physics. High temperatures lead to stochasticity and random exploration, selecting all actions almost equally likely regardless of their Q-values. In contrast, low temperatures lead to high exploitation of the Q-values, selecting the action with the highest Q-value with probability close to one. Intermediate values prefer actions proportionally to their relative competitiveness. In many applications, the temperature parameter is decreased over time, allowing initially high exploration and eventual exploitation of the knowledge encoded in the Q-values. Within the scope of this article, the temperature is kept constant for analytical simplicity and coherence with the derivations in [17, 18].

2.1.3 Replicator dynamics of learning automata

Using the example of learning automata, this section demonstrates the forward approach to modeling multi-agent reinforcement learning within a evolutionary game theoretic framework. In particular, we indicate the mathematical relation between learning automata and the multi-population replicator dynamics. For the full prove we refer to Börgers et al. [1].

The continuous time two-population replicator dynamics are defined by the following system of differential equations:

$$\begin{aligned} \frac{d\pi_i}{dt} &= \pi_i\Big[(A\sigma)_i - \pi' A\sigma\Big] \\ \frac{d\sigma_j}{dt} &= \sigma_j\Big[(B\pi)_j - \sigma' B\pi\Big] \end{aligned} \quad (2)$$

where A and B are the normal form game payoff matrices for player 1 and 2 respectively. The probability vector π describes the frequency of all pure strategies (replicators) for player 1. Success of a replicator i is measured by the difference between its current payoff $(A\sigma)_i$ and the average payoff $\pi' A\sigma$ of the entire population π against the strategy of player 2.

The policy change in (1) depends on action $a\,(t)$ selected at time t. We now assume that an agent receives an immediate reward for each possible action rather than just the feedback for this specific action $a(t)$. Furthermore, let the reward $\bar{r}_i$ for action i be the average reward that action i yields given that all other agents play according to their current policies. Finally, the action probability change in (1)

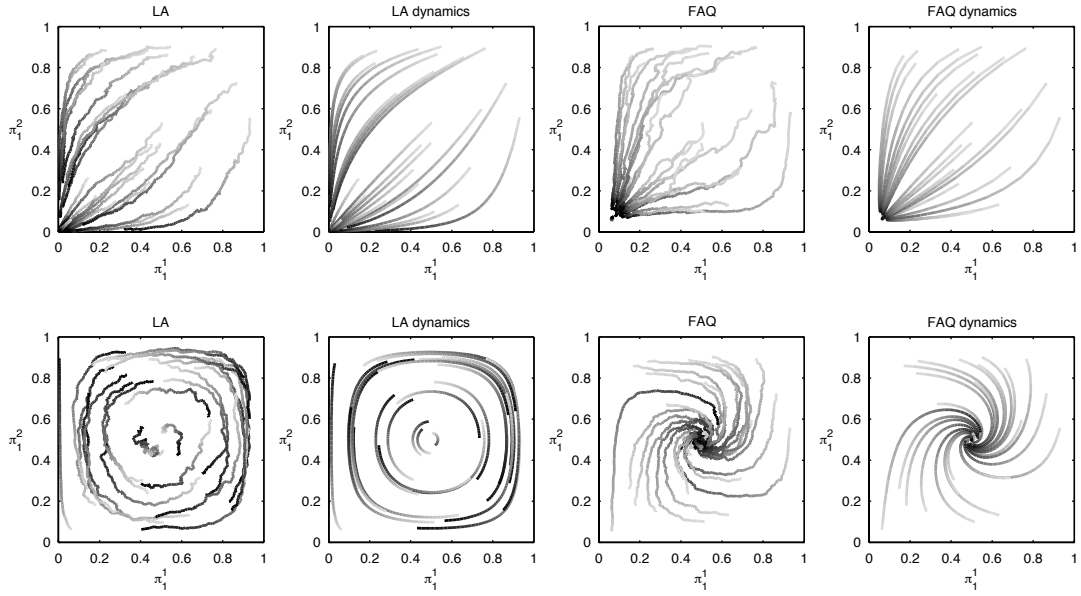

Figure 1: Overview of trajectory plots for stateless games: Prisoners' Dilemma (top row) and Matching Pennies game (bottom row).

is proportional to π_i since π_i determines the frequency of action i. Consequently, (3) describes the expected average policy change at time t.

$$\begin{aligned} E\left(\Delta\pi_i\left(t\right)\right) &= \pi_i\left[\alpha\bar{r}_i\left(t\right)\left(1-\pi_i\left(t\right)\right)+\sum_{j\neq i}\left(-\alpha r_j\left(t\right)\pi_j\left(t\right)\right)\right] \\ &= \pi_i\alpha\left[\bar{r}_i\left(t\right)-\bar{r}_i\left(t\right)\pi_i\left(t\right)-\sum_{j\neq i}\left(r_j\left(t\right)\pi_j\left(t\right)\right)\right] \\ &= \pi_i\alpha\left[\bar{r}_i\left(t\right)-\sum_{j}\left(r_j\left(t\right)\pi_j\left(t\right)\right)\right] \end{aligned} \tag{3}$$

If we apply (3) to a 2-player normal form game the connection between automata games and replicator dynamics becomes apparent. We consider a matrix game where A is the payoff for agent 1 and B the payoff for agent 2; π and σ are the two action probability distributions respectively. Agent 1 receives an average payoff of $\bar{r}_i = \left(A\sigma\right)_i$ for action i against agent 2's strategy σ. Hence, (3) can be rewritten as:

$$\begin{aligned} E\left(\Delta\pi_i\left(t\right)\right) &= \pi_i\alpha\left[\bar{r}_i\left(t\right)-\sum_{j}\left(r_j\left(t\right)\pi_j\left(t\right)\right)\right] \\ &= \pi_i\alpha\left[\left(A\sigma\right)_i-\sum_{j}\left(\left(A\sigma\right)_j\pi_j\right)\right] \\ &= \pi_i\alpha\Big[\left(A\sigma\right)_i-\pi' A\sigma\Big] \end{aligned} \tag{4}$$

Similarly, we can derive

$$E\left(\Delta\sigma_j\left(t\right)\right) = \sigma_j\alpha\Big[\left(B\pi\right)_j-\sigma' B\pi\Big] \tag{5}$$

for agent 2. Note that (4) and (5) correspond to the multi-population replicator equations given in (2) scaled by the learning rate α.

2.1.4 Dynamics of Q-learning

In [18] the authors extended the work of Borgers et al. [1] to Q-learning. More precisely, they derived the dynamics of the Q-learning process, which yields the following system of differential equations, describing the learning dynamics for a two-player stateless game:

$$\begin{aligned} \frac{d\pi_i}{dt} &= \pi_i\alpha\left(\tau^{-1}\Big[\left(A\sigma\right)_i-\pi' A\sigma\Big]-\log\pi_i+\sum_k\pi_k\log\pi_k\right) \\ \frac{d\sigma_j}{dt} &= \sigma_j\alpha\left(\tau^{-1}\Big[\left(B\pi\right)_j-\sigma' B\pi\Big]-\log\sigma_j+\sum_l\sigma_l\log\sigma_l\right) \end{aligned} \tag{6}$$

The equations contain a selection part, equal to the multi-population replicator dynamics, and a mutation part, originating from the Bolzmann exploration scheme of FAQ. For an elaborate discussion in terms of selection and mutation operators we refer to [17, 18].

2.1.5 Example single-state game analysis

We now examine the learning dynamics for a selection of 2 x 2 matrix games, in particular we consider the *Prisoners' Dilemma* and the *Matching Pennies* game. Reward matrices for Prisoners' Dilemma (left, *Defect* or *Cooperate*) and Matching Pennies (right, *Head* or *Tail*) are given below:

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

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

In all automata games the *linear reward-inaction* scheme with a reward parameter $\alpha = 0.005$ is used. Q-learners use a learning rate of $\alpha = 0.005$, discount factor $\gamma = 0$ and a constant temperature $\tau = 0.02$. Initial policies for learner and replicator trajectory plots are generated randomly.

Figure 1 (top row) shows the dynamics in the single state *Prisoners' Dilemma*. The automata game as well as the corresponding replicator dynamics show similar evolution toward the equilibrium strategy of mutual defection. Action probabilities are plotted for action 1 (in this case *cooperate*); x- and y-axis correspond to the action of player 1 and 2 respectively. Hence, the Nash equilibrium point is located at the origin $(0, 0)$. FAQ-learners evolve to a joint policy close to Nash. Constant temperature prohibits full convergence.

Learning in the *Matching Pennies* game, Figure 1 (bottom row), shows cyclic behavior for automata games and its replicator dynamics alike. FAQ-learning successfully converges to the mixed equilibrium due to its exploration scheme.

2.2 Multi-state learning dynamics

The main limitation of the evolutionary game theoretic approach to multi-agent learning has been its restriction to stateless repeated games. Even though real-life tasks might be modeled statelessly, the majority of such problems naturally relates to multi-state situations. Vrancx et al. [20] have made the first attempt to extend replicator dynamics to multi-state games. More precisely, the authors have combined replicator dynamics and piecewise dynamics, called piecewise replicator dynamics, to model the learning behavior of agents in stochastic games. Recently, this promising proof of concept has been formally studied in [5] and extended to *state-coupled replicator dynamics* [6] which form the foundation for the later described inverse approach.

2.2.1 Stochastic games

Stochastic games extend the concept of Markov decision processes to multiple agents, and allow to model multi-state games in an abstract manner. The concept of repeated games is generalized by introducing probabilistic switching between multiple states. At any time t, the game is in a specific state featuring a particular payoff function and an admissible action set for each player. Players take actions simultaneously and hereafter receive an immediate payoff depending on their joint action. A transition function maps the joint action space to a probability distribution over all states which in turn determines the probabilistic state change. Thus, similar to a Markov decision process, actions influence the state transitions. A formal definition of stochastic games (also called Markov games) is given below.

DEFINITION 1. *The game $G = \langle n, S, A, q, r, \pi^1 \dots \pi^n \rangle$ is a stochastic game with n players and k states. At each stage t, the game is in a state $s \in S = (s^1, \dots, s^k)$ and each player i chooses an action a^i from its admissible action set $A^i(s)$ according to its strategy $\pi^i(s)$.*

The payoff function $r(s,a) : \prod_{i=1}^{n} A^i(s) \mapsto \Re^n$ maps the joint action $a = (a^1, \dots, a^n)$ to an immediate payoff value for each player.

The transition function $q(s,a) : \prod_{i=1}^{n} A^i(s) \mapsto \Delta^{k-1}$ determines the probabilistic state change, where Δ^{k-1} is the $(k-1)$-simplex and $q_{s'}(s,a)$ is the transition probability from state s to s' under joint action a.

In this work we restrict our consideration to the set of games where all states $s \in S$ are in the same *ergodic set*. The motivation for this restriction is two-folded. In the presence of more than one ergodic set one could analyze the corresponding sub-games separately. Furthermore, the restriction ensures that the game has no absorbing states. Games with absorbing states are of no particular interest in respect to evolution or learning since any type of exploration will eventually lead to absorption. The formal definition of an ergodic set in stochastic games is given below.

DEFINITION 2. *In the context of a stochastic game G, $E \subseteq S$ is an ergodic set if and only if the following conditions hold:*
(a) For all $s \in E$, if G is in state s at stage t, then at $t+1$: $Pr(G \text{ in some state } s' \in E) = 1$, and
(b) for all proper subsets $E' \subset E$, (a) does not hold.

Note that in repeated games, player i either tries to maximize the limit of the average of stage rewards (e.g., Learning Automata)

$$\max_{\pi_i} \liminf_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r^i(t) \tag{7}$$

or the discounted sum of stage rewards $\sum_{t=1}^{T} r^i(t)\, \delta^{t-1}$ with $0 < \delta < 1$ (e.g., Q-learning), where $r^i(t)$ is the immediate stage reward for player i at time step t.

2.2.2 2-State Prisoners' Dilemma

The *2-State Prisoners' Dilemma* is a stochastic game for two players. The payoff matrices are given by

$$(A^1, B^1) = \begin{pmatrix} 3,3 & 0,10 \\ 10,0 & 2,2 \end{pmatrix}, (A^2, B^2) = \begin{pmatrix} 4,4 & 0,10 \\ 10,0 & 1,1 \end{pmatrix}.$$

Where A^s determines the payoff for player 1 and B^s for player 2 in state s. The first action of each player is *cooperate* and the second is *defect*. Player 1 receives $r^1(s,a) = A^s_{a_1,a_2}$ while player 2 gets $r^2(s,a) = B^s_{a_1,a_2}$ for a given joint action $a = (a_1, a_2)$. Similarly, the transition probabilities are given by the matrices $Q^{s \to s'}$ where $q_{s'}(s,a) = Q^{s \to s'}_{a_1,a_2}$ is the probability for a transition from state s to state s'.

$$Q^{s^1 \to s^2} = \begin{pmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{pmatrix},\ Q^{s^2 \to s^1} = \begin{pmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{pmatrix}$$

The probabilities to continue in the same state after the transition are $q_{s^1}(s^1, a) = Q^{s^1 \to s^1}_{a_1,a_2} = 1 - Q^{s^1 \to s^2}_{a_1,a_2}$ and $q_{s^2}(s^2, a) = Q^{s^2 \to s^2}_{a_1,a_2} = 1 - Q^{s^2 \to s^1}_{a_1,a_2}$.

Essentially a *Prisoners' Dilemma* is played in both states, and if regarded separately, *defect* is still a dominating strategy. One might assume that the Nash equilibrium strategy in this game is to *defect* at every stage. However, the only pure stationary equilibria in this game reflect strategies where one of the players *defects* in one state while *cooperating* in the other and the second player does exactly the opposite. Hence, a player betrays his opponent in one state while being exploited himself in the other state.

2.2.3 2-State Matching Pennies game

Another 2-player, 2-actions and 2-state game is the *2-State Matching Pennies* game. This game has a mixed Nash equilibrium with joint-strategies $\pi^1 = (.75, .25)$, $\pi^2 = (.5, .5)$ in state 1 and $\pi^1 = (.25, .75)$, $\pi^2 = (.5, .5)$ in state 2. Payoff and transition matrices are given below.

$$(A^1, B^1) = \begin{pmatrix} 1,0 & 0,1 \\ 0,1 & 1,0 \end{pmatrix}, (A^2, B^2) = \begin{pmatrix} 0,1 & 1,0 \\ 1,0 & 0,1 \end{pmatrix}$$

$$Q^{s^1 \to s^2} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix},\ Q^{s^2 \to s^1} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$$

2.2.4 Networks of learning automata

To cope with stochastic games, the learning algorithms in Section 2.1 need to be adopted to account for multiple states. To this end, we use a network of automata for each agent [19]. An agent associates a dedicated learning automaton (LA) to each state of the game and control is passed on from one automaton to another. Each LA tries to optimize the policy in its state using the standard update rule given in (1). Only a single LA is active and selects an action at each stage of the game. However, the immediate reward from the environment is not directly fed back to this LA. Instead, when the LA becomes active again, i.e., next time the same state is played, it is informed about the cumulative reward gathered since the last activation and the time that has passed by.

The reward feedback τ^i for agent i's automaton $\mathrm{LA}^i(s)$ associated with state s is defined as

$$\tau^i(t) = \frac{\Delta r^i}{\Delta t} = \frac{\sum_{l=t_0(s)}^{t-1} r^i(l)}{t - t_0(s)} , \tag{8}$$

where $r^i(t)$ is the immediate reward for agent i in epoch t and $t_0(s)$ is the last occurrence function and determines when states s was visited last. The reward feedback in epoch t equals the cumulative reward Δr^i divided by time-frame Δt. The cumulative reward Δr^i is the sum over all immediate rewards gathered in all states beginning with epoch $t_0(s)$ and including the last epoch $t-1$. The time-frame Δt measures the number of epochs that have passed since automaton $\mathrm{LA}^i(s)$ has been active last. This means the state policy is updated using the average stage reward over the interim immediate rewards.

2.2.5 Average reward game

For a repeated automata game, let the objective of player i at stage t_0 be to maximize the limit average reward $\bar{r}^i = \liminf_{T\to\infty} \frac{1}{T}\sum_{t=t_0}^{T} r^i(t)$ as defined in (7). The scope of this paper is restricted to stochastic games where the sequence of game states $X(t)$ is ergodic. Hence, there exists a stationary distribution x over all states, where fraction x_s determines the frequency of state s in X. Therefore, we can rewrite $\bar{r}^i$ as $\bar{r}^i = \sum_{s\in S} x_s P^i(s)$, where $P^i(s)$ is the expected payoff of player i in state s.

Now, let us assume the game is in state s at stage t_0 and players play a given joint action a in s and fixed strategies $\pi(s')$ in all states but s. Then the limit average payoff becomes

$$\bar{r}(s,a) = x_s r(s,a) + \sum_{s'\in S-\{s\}} x_{s'} P^i\left(s'\right) , \tag{9}$$

where

$$P^i\left(s'\right) = \sum_{a'\in\prod_{i=1}^n A^i(s')} \left(r\left(s',a'\right) \prod_{i=1}^n \pi^i_{a'_i}\left(s'\right) \right).$$

An intuitive explanation of (9) goes as follows. At each stage, players consider the infinite horizon of payoffs under current strategies. We untangle the current state s from all other states $s' \neq s$ and the limit average payoff $\bar{r}$ becomes the sum of the immediate payoff for joint action a in state s and the expected payoffs in all other states. Payoffs are weighted by the frequency x_s of corresponding state occurrences. Thus, if players invariably play joint action a every time the game is in state s and their fixed strategies $\pi(s')$ for all other states, the limit average reward for $T\to\infty$ is expressed by (9).

Since a specific joint action a is played in state s, the stationary distribution x depends on s and a as well. A formal definition is given below.

Definition 3. *For $G = \langle n, S, A, q, r, \pi^1 \dots \pi^n \rangle$ where S itself is the only ergodic set in $S = \left(s^1 \dots s^k\right)$, we say $x(s,a)$ is a stationary distribution of the stochastic game G if and only if $\sum_{z\in S} x_z(s,a) = 1$ and*

$$x_z(s,a) = x_s(s,a)\, q_z(s,a) + \sum_{s'\in S-\{s\}} x_{s'}(s,a)\, Q^i\left(s'\right),$$

where

$$Q^i\left(s'\right) = \sum_{a'\in\prod_{i=1}^n A^i(s')} \left(q_z\left(s',a'\right) \prod_{i=1}^n \pi^i_{a'_i}\left(s'\right) \right).$$

Based on this notion of stationary distribution and (9) we can define the average reward game as follows.

Definition 4. *For a stochastic game G where S itself is the only ergodic set in $S = \left(s^1 \dots s^k\right)$, we define the average reward game for some state $s \in S$ as the normal-form game*

$$\bar{G}\left(s, \pi^1 \dots \pi^n\right) = \left\langle n, A^1(s) \dots A^n(s), \bar{r}, \pi^1(s) \dots \pi^n(s) \right\rangle,$$

where each player i plays a fixed strategy $\pi^i(s')$ in all states $s' \neq s$. The payoff function $\bar{r}$ is given by

$$\bar{r}(s,a) = x_s(s,a)\, r(s,a) + \sum_{s'\in S-\{s\}} x_{s'}(s,a)\, P^i\left(s'\right).$$

2.2.6 State-coupled replicator dynamics

We reconsider the replicator equations for population π as given in (2):

$$\frac{d\pi_i}{dt} = \pi_i \left[(A\sigma)_i - \pi' A \sigma \right]$$

Essentially, the payoff of an individual in population π, playing pure strategy i against population σ, is compared to the average payoff of population π. In the context of an average reward game $\bar{G}$ with payoff function $\bar{r}$ the expected payoff for player i and pure action j is given by

$$P^i_j(s) = \sum_{a\in\prod_{l\neq i} A^l(s)} \left(\bar{r}^i(a^*) \prod_{l\neq i} \pi^l_{a^*_l}(s) \right),$$

where $a^* = \left(a^1 \dots a^{i-1}, j, a^i \dots a^n\right)$. This means that we enumerate all possible joint actions a with fixed action j for agent i. In general, for some mixed strategy ω, agent i receives an expected payoff of

$$P^i(s,\omega) = \sum_{j\in A^i(s)} \left[\omega_j \sum_{a\in\prod_{l\neq i} A^l(s)} \left(\bar{r}^i(s,a^*) \prod_{l\neq i} \pi^l_{a^*_l}(s) \right) \right].$$

If each player i is represented by a population π^i, we can set up a system of differential equations, each similar to (2), where the payoff matrix A is substituted by the average reward game payoff $\bar{r}$. Furthermore, σ now represents all remaining populations π^l where $l \neq i$.

Definition 5. *The multi-population state-coupled replicator dynamics are defined by the following system of differential equations:*

$$\frac{d\pi^i_j(s)}{dt} = \pi^i_j\; x_s(\pi) \left[P^i(s, e_j) - P^i\left(s, \pi^i(s)\right) \right], \tag{10}$$

where e_j is the j^{th}-unit vector. $P^i(s,\omega)$ is the expected payoff for an individual of population i playing some strategy ω in state s. P^i is defined as

$$P^i(s,\omega) = \sum_{j \in A^i(s)} \left[\omega_j \sum_{a \in \prod_{l \neq i} A^l(s)} \left(\bar{r}^i(s,a^*) \prod_{l \neq i} \pi^l_{a^*_l}(s) \right) \right],$$

where $\bar{r}$ is the payoff function of $\bar{G}\left(s, \pi^1 \dots \pi^n\right)$ and

$$a^* = \left(a^1 \dots a^{i-1}, j, a^i \dots a^n\right).$$

Furthermore, x is the stationary distribution over all states S under π, with

$$\sum_{s \in S} x_s(\pi) = 1 \text{ and}$$

$$x_s(\pi) = \sum_{z \in S} \left[x_z(\pi) \sum_{a \in \prod_{i=1}^n A^i(s)} \left(q_s(z,a) \prod_{i=1}^n \pi^i_{a_i}(s) \right) \right].$$

In total this system has $N = \sum_{s \in S} \sum_{i=1}^n |A^i(s)|$ replicator equations.

In essence, state-coupled replicator dynamics use direct state-coupling by incorporating the expected payoff in all states under current strategies, weighted by the frequency of state occurrences.

Previous work has shown that state-coupled replicator dynamics converge to pure Nash equilibria in general-sum stochastic games such as the 2-State Prisoners' Dilemma [6]. However, state-coupled replicator dynamics fail to converge to mixed equilibria. We observe cycling behavior, similar to the stateless situation of Matching Pennies (see Figure 2).

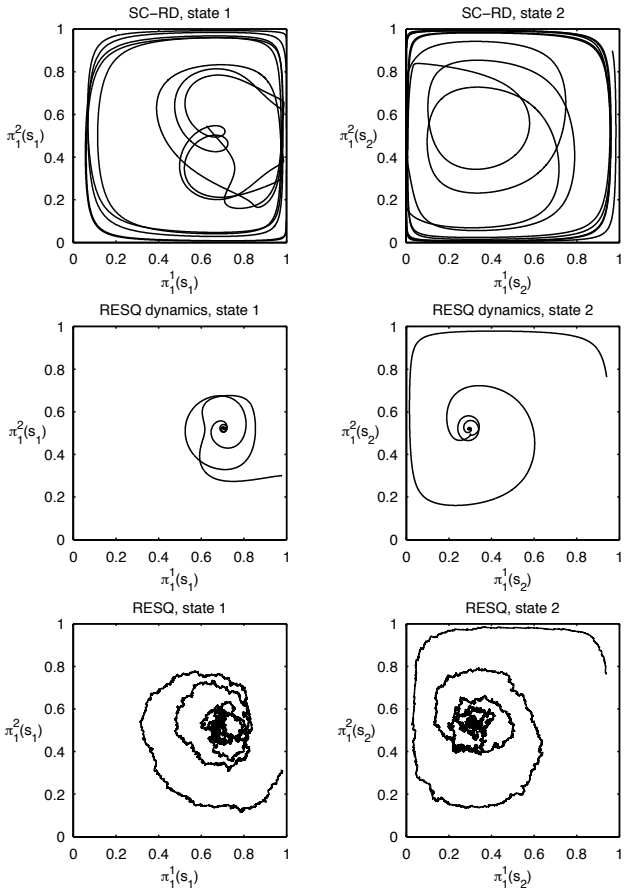

Figure 2: Comparison between SC-RD dynamics, RESQ dynamics and RESQ-learning ($\alpha = 0.004$, $\tau = 0.04$) in the 2-State Matching Pennies game.

3. INVERSE APPROACH

The forward approach has focused on deriving predictive models for the learning dynamics of existing multi-agent reinforcement learners. These models help to gain deeper insight and allow to tune parameter settings. In this section we demonstrate the inverse approach, designing a dynamical system that does indeed converge to pure and mixed Nash equilibria and reverse re-engineering that system, resulting in a new multi-agent reinforcement learning algorithm, i.e. RESQ-learning.

Results for stateless games provide evidence that exploration is the key to prevent cycling around attractors. Hence, we aim to combine the exploration-mutation term of FAQ-learning dynamics with state-coupled replicator dynamics.

3.1 Linking LA and Q-learning dynamics

First, we link the dynamics of learning automata and Q-learning for the stateless case. We recall from Section 2.1.3 that the learning dynamics of LA correspond to the standard multi-population replicators scaled by the learning rate α:

$$\frac{d\pi_i}{dt} = \pi_i \alpha \Big[(A\sigma)_i - \pi' A \sigma \Big]$$

The FAQ replicator dynamics (see Section 2.1.4) contain a selection part equivalent to the multi-population replicator dynamics, and an additional mutation part originating from the Bolzmann exploration scheme:

$$\begin{aligned} \frac{d\pi_i}{dt} &= \pi_i \beta \left(\tau^{-1} \Big[(A\sigma)_i - \pi' A \sigma \Big] - \log \pi_i + \sum_k \pi_k \log \pi_k \right) \\ &= \pi_i \beta \tau^{-1} \Big[(A\sigma)_i - \pi' A \sigma \Big] - \pi_i \beta \left(\log \pi_i + \sum_k \pi_k \log \pi_k \right) \end{aligned}$$

The learning rate of FAQ is now denoted by β. Let us assume $\alpha = \beta\tau^{-1} \Rightarrow \beta = \alpha\tau$. Note that from $\beta \in [0,1]$ follows

$$0 \leq \alpha\tau^{-1} \leq 1.$$

Then we can rewrite the FAQ replicator equation as follows:

$$\frac{d\pi_i}{dt} = \pi_i \alpha \Big[(A\sigma)_i - \pi' A \sigma \Big] - \pi_i \alpha\tau \left(\log \pi_i + \sum_k \pi_k \log \pi_k \right)$$

In the limit $\lim_{\tau \to 0}$ the mutation term collapses and the dynamics of learning automata become:

$$\frac{d\pi_i}{dt} = \pi_i \alpha \Big[(A\sigma)_i - \pi' A \sigma \Big]$$

3.2 State-coupled RD with mutation

After we have established the connection between the learning dynamics of FAQ-learning and learning automata, extending this link to multi-state games is straightforward.

The mutation term

$$-\tau\Big(\log\pi_i+\sum_k\pi_k\log\pi_k\Big) \tag{11}$$

is solely dependent on the agent's policy π and thus independent of any payoff computation. Therefore, the average reward game remains the sound measure for the limit of the average of stage rewards under the assumptions made in Section 2.2.5. The equations of the dynamical system in (2.2.5) are complemented with the mutation term (11), resulting in the following state-coupled replicator equations with mutation:

$$\frac{d\pi_j^i(s)}{dt}=\pi_j^i\;x_s(\pi)\Bigg[\Big[P^i(s,e_j)-P^i\Big(s,\pi^i(s)\Big)\Big] -\tau\Big(\log\pi_j^i+\sum_k\pi_k^i\log\pi_k^i\Big)\Bigg] \tag{12}$$

In the next section we introduces the corresponding RESQ-learning algorithm.

3.3 RESQ-learning

In [6] the authors have shown that maximizing the expected average stage reward over interim immediate rewards relates to the average reward game played in state-coupled replicator dynamics. We reverse this result to obtain a learner equivalent to state-coupled replicator dynamics with mutation.

Analog to the description in Section 2.2.4 a network of learners is used for each agent i. The reward feedback signal is equal to (8) while the update rule now incorporates the same exploration term as in (12). If $a(t)=i$:

$$\pi_i(t{+}1)\leftarrow\pi_i(t){+}\alpha\Bigg[r(t)\,(1{-}\pi_i(t)){-}\tau\Big(\log\pi_j^i+\sum_k\pi_k^i\log\pi_k^i\Big)\Bigg]$$

otherwise:

$$\pi_i(t{+}1)\leftarrow\pi_i(t){+}\alpha\Bigg[-r(t)\,\pi_i(t){-}\tau\Big(\log\pi_j^i+\sum_k\pi_k^i\log\pi_k^i\Big)\Bigg]$$

Hence, RESQ-learning is essentially a multi-state policy iterator using exploration equivalent to the Boltzmann policy generation scheme.

4. RESULTS AND DISCUSSION

This section sets the newly proposed RESQ-learning algorithm in perspective by examining the underlying dynamics of state-coupled replicator dynamics with mutation and traces of the resulting learning algorithm.

First, we explore the behavior of the dynamical system, as derived in Section 3.2, and verify the desired convergence behavior, i.e., convergence to pure and mixed Nash equilibria. Figure 3 shows multiple trajectory traces in the 2-State Prisoners' Dilemma, originating from random strategy profiles in both states. Analysis reveals that all trajectories converge close to either one of the two pure Nash equilib-

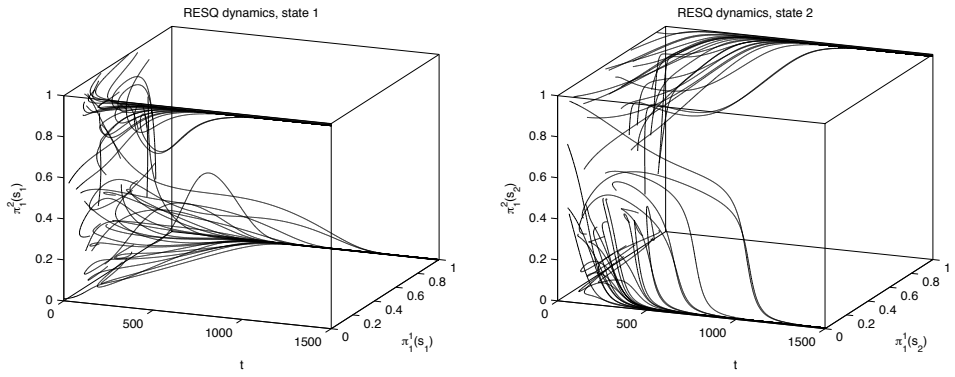

Figure 3: RESQ-learning dynamics ($\alpha=0.004$, $\tau=0.02$) in the 2-State Prisoners' Dilemma.

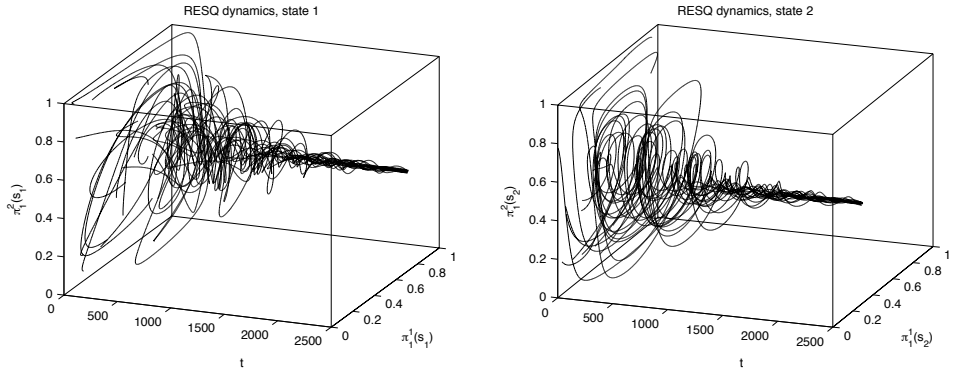

Figure 4: RESQ-learning ($\alpha=0.004$, $\tau=0.04$) in the 2-State Matching Pennies game.

rium points described in Section 2.2.2. As mentioned before for the stateless case, constant temperature prohibits full convergence. Figure 4 shows trajectory traces in the 2-State Matching Pennies game. Again, all traces converge close to Nash, thus affirming the statement that exploration-mutation is crucial to prevent cycling and to converge in games with mixed optimal strategies.

Figure 2 shows a comparison between state-coupled replicator dynamics (SC-RD), the RESQ-dynamics as in (12), and an empirical learning trace of RESQ-learners. As abovementioned, "pure" state-coupled replicator dynamics without the exploration-mutation term fail to converge. The trajectory of the state space of this dynamical system exhibits cycling behavior around the mixed Naish equilibrium (see Section 2.2.3). RESQ-dynamics successfully converge ϵ-near to the Nash-optimal joint policy. Furthermore, we present the learning trace of two RESQ-learners in order to judge the predictive quality of the coresponding state-coupled dynamics with mutation. Due to the stochasticity involved in the action selection process, the learning trace is more noisy. However, we clearly observe that RESQ-learning indeed successfully inherits the convergence behavior of state-coupled replicator dynamics with mutation.

Further experiments are required to verify the performance of RESQ-learning in real applications and to gain insight into how it competes with multi-state Q-learning and the SARSA algorithm [15]. In particular, the speed and quality of convergence need to be considered. Therefore, the theoretical framework needs to be extended to account for decreasing temperature to balance exploration and exploitation over time.

5. CONCLUSIONS

The contributions of this article can be summarized as follows. First, we have demonstrated the forward approach to modeling multi-agent reinforcement learning within an evolutionary game theoretic framework. In particular, the stateless learning dynamics of learning automata and FAQ-learning as well as state-coupled replicator dynamics for stochastic games have been discussed. Based on the insights that were gained from the forward approach, RESQ-learning has been introduced by reverse engineering state-coupled replicator dynamics injected with the Q-learning Boltzmann mutation scheme. We have provided empirical confirmation that RESQ-learning successfully inherits the convergence behavior of its evolutionary counter part. Results have shown that RESQ-learning provides convergence to pure as well as mixed Nash equilibria in a selection of stateless and stochastic multi-agent games.

6. REFERENCES

[1] Tilman Börgers and Rajiv Sarin. Learning through reinforcement and replicator dynamics. *Journal of Econ. Theory*, 77(1), 1997.

[2] Bruce Bueno de Mesquita. Game theory, political economy, and the evolving study of war and peace. *American Political Science Review*, 100(4):637–642, November 2006.

[3] Herbert Gintis. *Game Theory Evolving. A Problem-Centered Introduction to Modelling Strategic Interaction*. Princeton University Press, Princeton, 2000.

[4] Eduardo Rodrigues Gomes and Ryszard Kowalczyk. Dynamic analysis of multiagent q-learning with epsilon-greedy exploration. In *ICML*, 2009.

[5] Daniel Hennes, Karl Tuyls, and Matthias Rauterberg. Formalizing multi-state learning dynamics. In *Proc. of 2009 Intl. Conf. on Intelligent Agent Technology*, 2008.

[6] Daniel Hennes, Karl Tuyls, and Matthias Rauterberg. State-coupled replicator dynamics. In *Proc. of 8th Intl. Conf. on Autonomous Agents and Multiagent Systems*, 2009.

[7] Shlomit Hon-Snir, Dov Monderer, and Aner Sela. A learning approach to auctions. *Journal of Economic Theory*, 82:65–88, November 1998.

[8] Junling Hu and Michael P. Wellman. Nash q-learning for general-sum stochastic games. *Journal of Machine Learning*, 4:1039–1069, 2003.

[9] Michael Kaisers and Karl Tuyls. Frequency adjusted multi-agent q-learning. In *Proc. of 9th Intl. Conf. on Autonomous Agents and Multiagent Systems*, 2010.

[10] Michael L. Littman. Friend-or-foe q-learning in general-sum games. In *ICML*, pages 322–328, 2001.

[11] Shervin Nouyan, Roderich Groß, Michael Bonani, Francesco Mondada, and Marco Dorigo. Teamwork in self-organized robot colonies. *Transactions on Evolutionary Computation*, 13(4):695–711, 2009.

[12] Liviu Panait, Karl Tuyls, and Sean Luke. Theoretical advantages of lenient learners: An evolutionary game theoretic perspective. *Journal of Machine Learning Research*, 9:423–457, 2008.

[13] S. Phelps, M. Marcinkiewicz, and S. Parsons. A novel method for automatic strategy acquisition in n-player non-zero-sum games. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 705–712, Hakodate, Japan, 2006. ACM.

[14] Y. Shoham, R. Powers, and T. Grenager. If multi-agent learning is the answer, what is the question? *Journal of Artificial Intelligence*, 171(7):365–377, 2006.

[15] Richard S. Sutton and Aandrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[16] K. Tuyls and S. Parsons. What evolutionary game theory tells us about multiagent learning. *Artificial Intelligence*, 171(7):115–153, 2007.

[17] Karl Tuyls, Pieter J. 't Hoen, and Bram Vanschoenwinkel. An evolutionary dynamical analysis of multi-agent learning in iterated games. *Autonomous Agents and Multi-Agent Systems*, 12:115–153, 2005.

[18] Karl Tuyls, Katja Verbeeck, and Tom Lenaerts. A selection-mutation model for Q-learning in multi-agent systems. In *Proc. of 2nd Intl. Conf. on Autonomous Agents and Multiagent Systems*, 2003.

[19] Katja Verbeeck, Peter Vrancx, and Ann Nowé. Networks of learning automata and limiting games. In *ALAMAS*, 2006.

[20] Peter Vrancx, Karl Tuyls, Ronald Westra, and Ann Nowé. Switching dynamics of multi-agent learning. In *Proc. of 7th Intl. Conf. on Autonomous Agents and Multiagent Systems*, 2008.