State-Coupled Replicator Dynamics

Daniel Hennes
Eindhoven University of Technology
P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
d.hennes@tue.nl

Karl Tuyls
Eindhoven University of Technology
P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
k.p.tuyls@tue.nl

Matthias Rauterberg
Eindhoven University of Technology
P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
g.w.m.rauterberg@tue.nl

ABSTRACT

This paper introduces a new model, i.e. state-coupled replicator dynamics, expanding the link between evolutionary game theory and multiagent reinforcement learning to multi-state games. More precisely, it extends and improves previous work on piecewise replicator dynamics, a combination of replicators and piecewise models. The contributions of the paper are twofold. One, we identify and explain the major shortcomings of piecewise replicators, i.e. discontinuities and occurrences of qualitative anomalies. Two, this analysis leads to the proposal of the new model for learning dynamics in stochastic games, named state-coupled replicator dynamics. The preceding formalization of piecewise replicators - general in the number of agents and states - is factored into the new approach. Finally, we deliver a comparative study of finite action-set learning automata to piecewise and state-coupled replicator dynamics. Results show that state-coupled replicators model learning dynamics in stochastic games more accurately than their predecessor, the piecewise approach.

Categories and Subject Descriptors

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

General Terms

Algorithms, Theory

Keywords

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

1. INTRODUCTION

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 [5]. 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, an evolutionary game theoretic approach has been employed to fill this gap [6]. In particular, in [1] the authors have derived a formal relation between multi-agent reinforcement learning and the replicator dynamics. The relation between replicators and reinforcement learning has been extended to different algorithms such as learning automata and Q-learning in [7].

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 [4].

The main limitation of this game theoretic approach to multi-agent systems is 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. [9] 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, we have formalized this promising proof of concept in [2].

Piecewise models are a methodology in the area of dynamical system theory. The core concept is to partition the state space of a dynamical system into cells. The behavior of a dynamical system can then be described as the state vector movement through this collection of cells. Dynamics within each cell are determined by the presence of an attractor or repeller. Piecewise linear systems make the assumption that each cell is reigned by a specific attractor and that the induced dynamics can be approximated linearly.

In this work, we demonstrate the major shortcomings of piecewise modeling in the domain of replicator dynamics and subsequently propose a new method, named *state-coupled replicator dynamics*. Grounded on the formalization of piecewise replicators, the new model describes the direct coupling between states and thus overcomes the problem of anomalies induced by approximation.

Cite as: State-Coupled Replicator Dynamics, Daniel Hennes, Karl Tuyls, Matthias Rauterberg, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 789–796

The rest of this article is organized as follows. Section 2 provides background information about the game theoretical framework and the theory of learning automata. In Section 3 we formally introduce piecewise replicator dynamics and hereafter discuss their shortcomings in Section 4. Section 5 presents the new state-coupled model and delivers a comparative study of piecewise and state-coupled replicator dynamics to learning automata. Section 6 concludes this article.

2. BACKGROUND

In this section, we summarize required background knowledge from the fields of multi-agent learning and evolutionary game theory. In particular, we consider an individual level of analogy between the related concepts of learning and evolution. Each agent has a set of possible strategies at hand. Which strategies are favored over others depends on the experience the agent has previously gathered by interacting with the environment and other agents. The pool of possible strategies can be interpreted as a population in an evolutionary game theory perspective. The dynamical change of preferences within the set of strategies can be seen as the evolution of this population as described by the replicator dynamics (Section 2.1). We leverage the theoretical framework of stochastic games (Section 2.2) to model this learning process and use learning automata as an example for reinforcement learning (Section 2.3).

2.1 Replicator dynamics

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

$$\begin{aligned}\frac{d\pi_i}{dt} &= \left[(A\sigma)_i - \pi' A\sigma\right]\pi_i \\ \frac{d\sigma_i}{dt} &= \left[(B\pi)_i - \sigma' B\pi\right]\sigma_i \ ,\end{aligned} \tag{1}$$

where A and B are the 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 of the entire population π against the strategy of player 2: $\pi' A\sigma$.

2.2 Stochastic games

Stochastic games allow to model multi-state problems in an abstract manner. The concept of repeated games is generalized by introducing probabilistic switching between multiple states. In each stage, 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 $G = \langle n, S, A, q, \tau, \pi^1 \dots \pi^n \rangle$ is a stochastic game with n players and k states. In each state $s \in S = (s^1,\dots,s^k)$ 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 $\tau(s,a) : \prod_{i=1}^n A^i(s) \mapsto \Re^n$ 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

$$\max_{\pi_i} \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \tau^i(t) \tag{2}$$

or the discounted sum of stage rewards $\sum_{t=1}^{T} \tau^i(t)\,\delta^{t-1}$ with $0 < \delta < 1$, where $\tau^i(t)$ is the immediate stage reward for player i at time step t. While the latter is commonly used in Q-learning, this work regards the former to derive a temporal difference reward update for learning automata in Section 2.3.1.

2.2.1 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 $\tau^1(s,a) = A^s_{a_1,a_2}$ while player 2 gets $\tau^2(s,a) = B^s_{a_1,a_2}$ 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.

2.2.2 Common Interest Game

Another 2-player, 2-actions and 2-state game is the *Common Interest Game*. Payoff and transition matrices are given below. Note that both players receive identical immediate rewards.

$$A^1 = B^1 = \begin{pmatrix} 5 & 6 \\ 6 & 7 \end{pmatrix},\ A^2 = B^2 = \begin{pmatrix} 0 & 10 \\ 5 & 0 \end{pmatrix}$$

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

In this game the highest payoff is gained in state 2 under joint action $(1, 2)$ which is associate with a low state transition probability $Q_{1,2}^{s^2 \to s1} = 0.1$. If however the state is switched, players are encouraged to play either joint action $(1, 2)$ or $(2, 1)$ in order to transition back to state 2 with a high probability. While joint action $(2, 2)$ maximizes immediate payoff in state 1, the low transition probability $Q_{2,2}^{s^1 \to s^2} = 0.1$ hinders the return to state 2.

2.3 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.

In this paper we focus on *finite action-set learning automata* (FALA). FALA are 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. Convergence for learning automata in single and specific multi-agent cases has been proven in [3].

The class of finite action-set learning automata considers only automata that optimize their policies over a finite action-set $A = \{1, \ldots, k\}$ 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 distribution $\pi(t)$, called policy. Based on the action $a(t)$, the environment responds with a reinforcement signal $\tau(t)$, called reward. Hereafter, the automaton uses the reward $\tau(t)$ to update $\pi(t)$ to the new policy $\pi(t+1)$. The update rule for FALA using the *linear reward-inaction (L_{R-I})* scheme is given below.

$$\pi_i(t+1) = \pi_i(t) + \begin{cases} \alpha\tau\left(t\right)\left(1 - \pi_i(t)\right) & \text{if } a\left(t\right) = i \\ -\alpha\tau\left(t\right)\pi_i(t) & \text{otherwise} \end{cases}$$

where $\tau \in [0, 1]$. The reward parameter $\alpha \in [0, 1]$ determines the learning rate.

Situating automata in stateless games is straightforward and only a matter of unifying the different taxonomies of game theory and the theory of learning automata (e.g. "player" and "agent" are interchangeable, as are "payoff" and "reward"). However, multi-state games require an extension of the stateless FALA model.

2.3.1 Networks of learning automata

For each agent, we use a network of automata in which control is passed on from one automaton to another [8]. An agent associates a dedicated learning automata to each state of the game. This LA tries to optimize the policy in that state using the standard update rule given in (2.3). 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

$$\tau^i\left(t\right) = \frac{\Delta r^i}{\Delta t} = \frac{\sum_{l=t_0(s)}^{t-1} r^i\left(l\right)}{t - t_0(s)}\ , \tag{3}$$

where $r^i\left(t\right)$ 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.

3. PIECEWISE REPLICATOR DYNAMICS

As outlined in the previous section, agents maintain an independent policy for each state and this consequently leads to a very high dimensional problem. *Piecewise replicator dynamics* analyze the dynamics per state in order to cope with this problem. For each state of a stochastic game a so-called *average reward game* is derived. An average reward game determines the expected reward for each joint action in a given state, assuming fixed strategies in all other states. This method projects the limit average reward of a stochastic game onto a stateless normal-form game which can be analyzed using the multi-population replicator dynamics given in (1).

In general we can not assume that strategies are fixed in all but one state. Agents adopt their policies in all states in parallel and therefore the average reward game along with the linked replicator dynamics are changing as well. The core idea of piecewise replicator dynamics is to partition the strategy space into cells, where each cell corresponds to a set of attractors in the average reward game. This approach is based on the methodology of piecewise dynamical systems.

In dynamic system theory, the state vector of a system eventually enters an area of attraction and becomes subject to the influence of this attractor. In case of piecewise replicator dynamics the state vector is an element of the strategy space and attractors resemble equilibrium points in the average reward game. It is assumed that the dynamics in each cell are reigned by a set of equilibria and therefore we can qualitatively describe the dynamics of each cell by a set of replicator equations.

We use this approach to model learning dynamics in stochastic games as follows. For each state of a stochastic game we derive the average reward game (Section 3.1) and consider the strategy space over all joint actions for all other states. This strategy space is then partitioned into cells (Section 3.2), where each cell corresponds to a set of equilibrium points in the average reward game. We sample the strategy space of each cell (Section 3.3) and compute the corresponding limit average reward for each joint action, eventually leading to a set of replicator equations for each cell (Section 3.4).

More precisely, each state features a number of cells, each related to a set of replicator dynamics. For each state, a single cell is active and the associated replicator equations determine the dynamics in that state, while the active cell of a particular state is exclusively determined by the strategies in all other states. Strategy changes occur in all states in parallel and hence mutually induce cell switching.

3.1 Average reward game

For a repeated automata game, the objective of player i at stage t_0 is to maximize the limit average reward $\bar{\tau}^i = \liminf_{T\to\infty} \frac{1}{T} \sum_{t=t_0}^{T} \tau^i(t)$ as defined in (2). The scope of this paper is restricted to stochastic games where the sequence of game states $X(t)$ is ergodic (see Section 2.2). 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 $\bar{\tau}^i$ as $\bar{\tau}^i = \sum_{s\in S} x_s P^i(s)$, where $P^i(s)$ is the expected payoff of player i in state s.

In piecewise replicator dynamics, states are analyzed separately to cope with the high dimensionality. Thus, 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

$$\bar{\tau}(s,a) = x_s \tau(s,a) + \sum_{s'\in S-\{s\}} x_{s'} P^i\left(s'\right) \quad , \qquad (4)$$

where

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

An intuitive explanation of (4) 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 $\bar{\tau}$ 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 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 (4).

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 $G = \langle n, S, A, q, \tau, \pi^1 \dots \pi^n \rangle$ 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 (4) 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*

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

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

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

This formalization of average reward games has laid the basis for the definition and analysis of pure equilibrium cells.

3.2 Equilibrium cells

The average reward game projects the limit average reward for a given state onto a stateless normal-form game. This projection depends on the fixed strategies in all other states. In this section we explain how this strategy space can be partitioned into discrete cells, each corresponding to a set of equilibrium points of the average reward game.

First, we introduce the concept of a *pure equilibrium cell*. Such a cell is a subset of all strategy profiles under which a given joint action specifies a pure equilibrium in the average reward game. In a Nash equilibrium situation, no player can improve its payoff by unilateral deviation from its own strategy π^i. In the context of an average reward game, all strategies including $\pi^i(s')$ are fixed for all states $s' \neq s$. Therefore, the payoff $\bar{\tau}^i(s,a)$ (see (4)) depends only on the joint action a in state s. Hence, the equilibrium constraint translates to:

$$\forall_i \forall_{a^{i\prime}\in A^i(s)} : \bar{\tau}^i(s,a) \geq \bar{\tau}^i\left(s, a^i \dots a^{i-1}, a^{i\prime}, a^{i+1} \dots a^n\right)$$

Consequently, this leads to the following definition of pure equilibrium cells.

Definition 5. *We call $C(s,a)$ a pure equilibrium cell of a stochastic game G if and only if $C(s,a)$ is the subset of all strategy profiles $\pi = \left(\pi^1 \dots \pi^n\right)$ under which the following condition holds*

$$\forall_i \forall_{a'} : \bar{\tau}^i(s,a) \geq \bar{\tau}^i\left(s,a'\right),$$

where $\bar{\tau}$ is the payoff function of the average reward game $\bar{G}\left(s, \pi^1 \dots \pi^n\right)$; a and a' are joint actions where $\forall_{j\neq i} : a'^j = a^j$. Thus, a is a pure equilibrium in state s for all strategy profiles $\pi \in C(s,a)$.

Note that $\bar{\tau}$ is independent of the players' strategies in s. Hence, we can express the cell boundaries in state $s = s^i$ as a function of the profiles $\pi\left(s^1\right) \dots \pi\left(s^{i-1}\right), \pi\left(s^{i+1}\right) \dots \pi\left(s^k\right)$, i.e. players' strategies in all but state s. However, pure equilibrium cells might very well overlap for certain areas of this strategy space [9]. Therefore, we consider all possible combinations of equilibrium points within one state and partition the strategy space of all other states into corresponding discrete cells.

3.3 Strategy space sampling

The partitioned strategy space is sampled in order to compute particular average reward game payoffs that in turn are used to obtain the set of replicator equations. For each state and each discrete cell, the corresponding strategy space is

scanned using an equally spaced grid. Each grid point defines a specific joint strategy of all states but the one under consideration. Average reward game payoffs are averaged over all grid points and the resulting payoffs are embedded in the set of replicator equations RD_{c_s} for the specified cell c and state s.

3.4 Definition

This section links average reward game, pure equilibrium cells and strategy space sampling in order to obtain a coherent definition of piecewise replicator dynamics.

For each state $s = s^i$ the strategy space in all remaining states is partitioned into discrete cells. Each cell $c_s \subset A\left(s^1\right) \times \ldots \times A\left(s^{i-1}\right) \times A\left(s^{i+1}\right) \times \ldots \times A\left(s^k\right)$ refers to some combination of pure equilibria. This combination might as well resemble only a single equilibrium point or the empty set, i.e. no pure equilibrium in the average reward game. As explained in the previous section, the strategy subspace of each cell is sampled. As a result, we obtain payoff matrices which in turn lead to a set of replicator equations RD_{c_s} for each cell. However, the limiting distribution over states under the strategy π has to be factored into the system. This means that different strategies result in situations where certain states are played more frequently than others. Since we model each cell in each state with a separate set of replicator dynamics, we need to scale the change of $\pi(s)$ with frequency x_s. The frequency x_s determines the expected fraction of upcoming stages played in state s.

Definition 6. *The piecewise replicator dynamics are defined by the following system of differential equations:*

$$\frac{d\pi(s)}{dt} = RD_{c_s}(\pi(s))\, x_s,$$

where c_s is the active cell in state s and RD_{c_s} is the set of replicator equations that reign in cell c_s. Furthermore, x is the stationary distribution over all states S under π, with $\sum_{s \in S} x_s(\pi) = 1$ and

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

Note that x_s is defined similarly to Definition 3. However, here x_s is independent of joint action a in s but rather assumes strategy $\pi(s)$ to be played instead.

4. ANOMALIES OF PIECEWISE REPLICATOR DYNAMICS

This section shows the shortcomings of piecewise replicators by comparing the dynamics of learning automata to predictions from the piecewise model. The layered sequence plot in Figure 1 is used to observe and describe the different dynamics. Each still image consists of three layers, the learning trace of automata (L_{R-I} with $a = 0.001$), cell partitioning and a vector field. The learning trace in state 1 are plotted together with the cell boundaries for state 2 and vice versa. Depending on the current end-point location of the trajectory in state 1 we illustrate the dynamics in state 2 by plotting the vector field of the corresponding set of replicator equations. For the particular example in Figure 1, the trajectory in state 2 does not cross any cell boundaries and therefore the vector field in state 1 remains unchanged during the sequence. The learning trace in

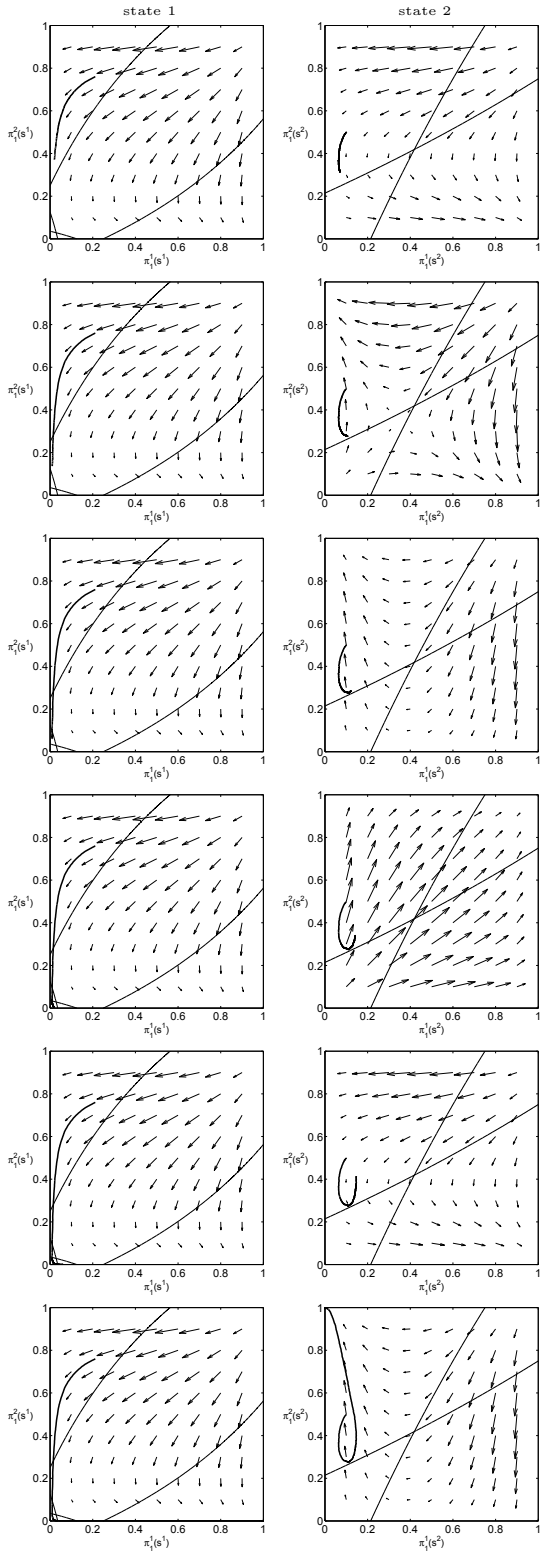

Figure 1: Trajectory plot for learning automata in the *2-State Prisoners' Dilemma*. Each boundary intersection in state 1 (left column) causes a qualitative change of dynamics in state 2 (right column).

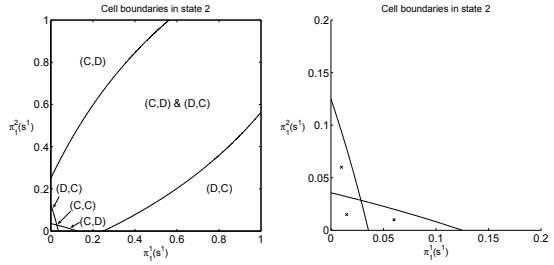

Figure 2: Cell partitioning and location of specific strategy profiles, i.e. $(0.015, 0.015)$**,** $(0.06, 0.01)$ **and** $(0.01, 0.06)$**, for vector field sensitivity analysis in the *2-State Prisoners' Dilemma*.**

state 1 follows this vector field and eventually converges to joint action *cooperate-defect* (C, D) resembling point $(1, 0)$ in the strategy space. During this process, the trajectory intersects multiple cell boundaries and the vector field in state 2 changes accordingly. We consult Figure 2 to identify cell labels for the partitioning plotted in the strategy space of state 1.

The first row of Figure 1 shows that the current end-point of the learning trajectory in state 1 is within the boundaries of cell (C, D). This means, cell (C, D) is active in state 2 and the dynamics are reigned by this attractor. In fact, we see that the policies of learning automata are attracted by this equilibrium point and approximately follow the vector field toward $(1, 0)$.

Let us now consider the third and fifth row. Here, the current end-points of the trajectories in state 1 correspond to the cells (D, C) and (C, D) respectively. In the former case, the vector field plot shows arrows near the end of the trace that point toward $(0, 1)$. However, the automata policies continue on an elliptic curve and therefore approximately progress into the direction of $(1, 1)$ rather than $(0, 1)$. The latter case shows even greater discrepancies between vector field and policy trajectory. The vector field predicts movement toward $(0, 0)$, while the trajectory trace continues convergence to $(0, 1)$. We now attempt to given an explanation for these artifacts by performing a sensitivity analysis of vector field plots.

Figure 2 displays the strategy space partitioning for state 2 depending on strategies in state 1 (left) as well as a magnification of the subspace near the origin (right). We specifically focus on this subspace and compute the average reward games for three strategy profiles, i.e. $(0.015, 0.015)$, $(0.06, 0.01)$ and $(0.01, 0.06)$. Each strategy profile corresponds to one of three cells, (C, C), (C, D) and (D, C) respectively, as indicated in the clipped section to the right in Figure 2. The computed average reward game payoff matrices are used to derive the replicator dynamics and visualize the vector field plots. Figure 3 compares the field plots of the three specific average reward games with the dynamics for the corresponding cells obtained by sampling. On the highest level, i.e. presence and convergence to strong attractors, all pairs match. This is clear, since average reward games for specific points within a cell sustain the same equilibrium combination. However, in order to examine the anomaly in Figure 1 (fifth still image), we consider especially the direction of field plots in the area around the trajectory end-point, which in this case is circa $(0.15, 0.4)$. In this area dynamics for cells (C, D) and (D, C) show clear qualitative differences from the field plots for corresponding specific strategy profiles $(0.06, 0.01)$ and $(0.01, 0.06)$. Furthermore, the field plots to the left in Figure 3 are consistent with the learning trajectory sequence displayed in Figure 1.

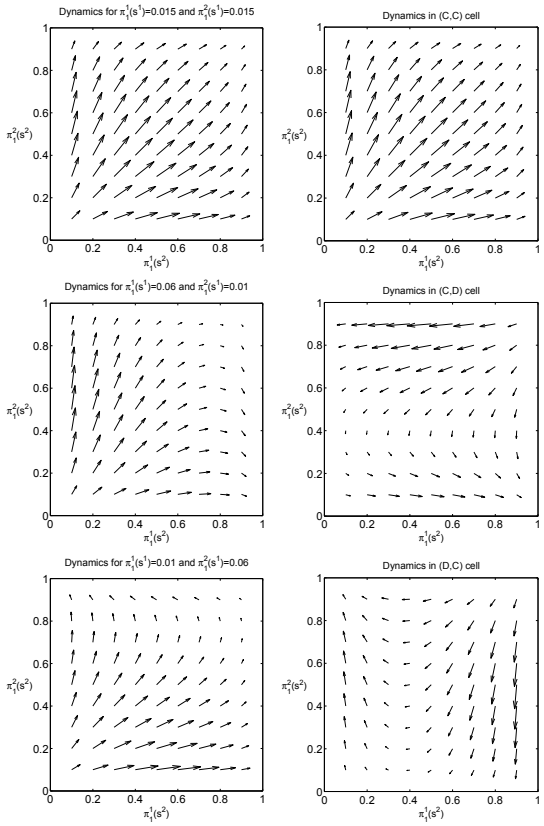

Figure 3: Vector field sensitivity analysis for piecewise replicator dynamics. Cell replicator equations are compared to the dynamics of specific average reward games for the following strategy profiles: $(0.015, 0.015)$**,** $(0.06, 0.01)$ **and** $(0.01, 0.06)$**.**

The vector field plots of piecewise replicator dynamics predict the learning dynamics only on the highest level (see Figure 3). However, due to the coupling of states, qualitative anomalies occur. Furthermore, piecewise replicators show discontinuities due to discrete switching of dynamics each time a cell boundary is crossed. These discontinuities are neither intuitive nor reflected in real learning behavior. Additionally, these artifacts potentially yield profound impact if the number of states is increased.

The next section aims to alleviate the shortcomings of piecewise replicators by proposing a new approach to model the learning dynamics in stochastic games more accurately.

5. STATE-COUPLED REPLICATORS

Piecewise replicators are an implementation of the paradigm of piecewise dynamical systems and therefore inherently limited to a qualitative approximation of the underlying, intrinsic behavior. Anomalies occur if this approximation is deficient for some area in the strategy space. This observation directly suggest either a) to refine the cell partitioning or b) to strive for direct state coupling, discarding the piecewise model.

Refining the cell partitioning is not straightforward. One

might consider to separate the disjointed parts of the strategy subspace, previously covered by a single cell. More precisely, this would lead to two separate cells each for pure equilibria (C, D) and (D, C) in state 2 of the *2-State Prisoners' Dilemma*. However, this approach is not in line with the argumentation that a cell should reflect the subspace in which a certain attractor reigns.

The second option is most promising since it eliminates undesired discontinuities induced by discrete cell switching and furthermore avoids approximation anomalies. Accordingly, the next section derives a new model for state-coupled learning dynamics in stochastic games. We call this approach *state-coupled replicator dynamics*.

5.1 Definition

We reconsider the replicator equations for population π as given in (1):

$$\frac{d\pi_i}{dt} = \left[(A\sigma)_i - \pi' A\sigma\right] \pi_i \tag{5}$$

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 $\bar{\tau}$ the expected payoff for player i and pure action j is given by

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

where $a = (a_1 \dots a_{i-1}, j, a_i \dots a_n)$. 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

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

If each player i is represented by a population π^i, we can set up a system of differential equations, each similar to (5), where the payoff matrix A is substituted by the average reward game payoff $\bar{\tau}$. Furthermore, σ now represents all remaining populations π^l where $l \neq i$.

Definition 7. *The multi-population state-coupled replicator dynamics are defined by the following system of differential equations:*

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

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

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

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

$$a = (a_1 \dots a_{i-1}, j, a_i \dots a_n).$$

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

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

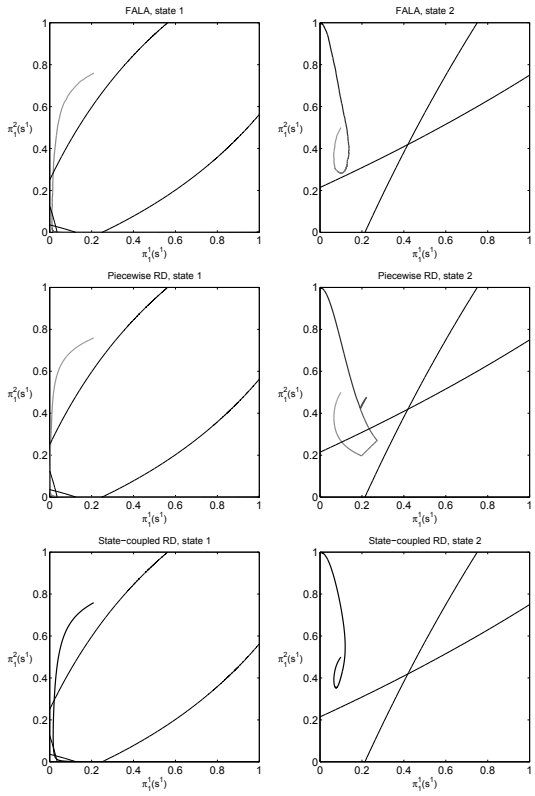

Figure 4: Comparison between a single trajectory trace of learning automata, piecewise and state-coupled replicator dynamics in the *2-State Prisoners' Dilemma*. The piecewise replicator dynamics clearly show discontinuities, while the state-coupled replicators model the learning dynamics more accurately.

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

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

Piecewise replicator dynamics rely on a cell partitioning, where the dynamics in each cell are approximated by a static set of replicator equations. In contrast, the 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.

5.2 Results and discussion

This section sets the newly proposed state-coupled replicator dynamics in perspective by comparing their dynamics with learning automata and piecewise replicators.

Figure 4 plots a single trace each for learning automata as well as piecewise and state-coupled replicator dynamics in the *2-State Prisoners' Dilemma*. All three trajectories converge to the same equilibrium point. The piecewise replicator dynamics clearly show discontinuities due to switching dynamics, triggered at each cell boundary intersection. Furthermore, the trace in state 2 enters the subspace for cell (D, D) in state 1, while both trajectories for learning automata and state-coupled replicators remain in cell (C, D).

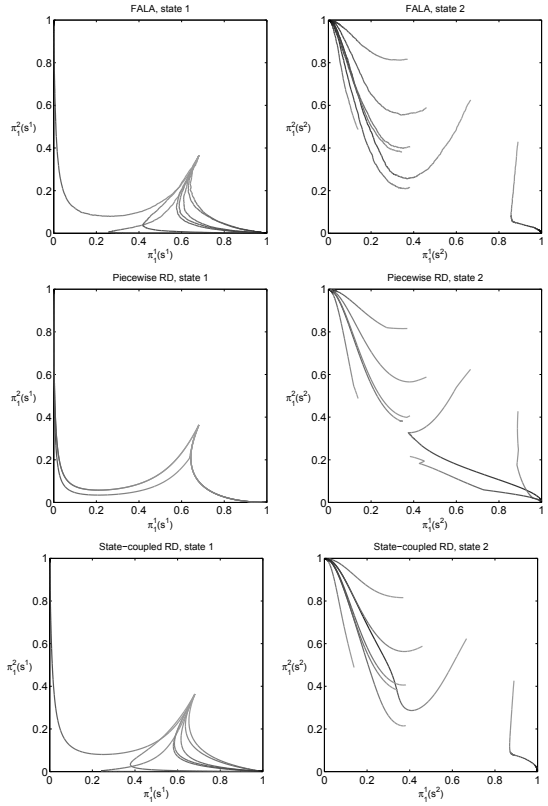

Figure 5: Comparison between trajectory traces of learning automata, piecewise and state-coupled replicator dynamics in the *2-State Prisoners' Dilemma*. Initial action probabilities are fixed in state 1 while a set of 8 random strategy profiles is used in state 2. Wrong convergence and discontinuities are observed for piecewise replicators.

This comparison clearly shows that state-coupled replicators model the learning dynamics more precisely.

In Figure 5 we compare multiple trajectory traces originating from one fixed strategy profile in state 1 and a set of randomly chosen strategies in state 2. This allows to judge the predictive quality of piecewise- and state-coupled replicator dynamics with respect to the learning curves of automata games.

Finally, Figure 6 presents trajectory plots for the *Common Interest Game*. Automata games and learning dynamics modeled by state-coupled replicator dynamics are compared. The strong alignment between model and real learning traces are evident in this game just as for the *2-State Prisoners' Dilemma*.

The new proposed state-coupled replicator dynamics directly describe the coupling between states and hence no longer rely on an additional layer of abstraction like piecewise cell dynamics. We observe in a variety of results that state-coupled replicator dynamics model multi-agent reinforcement learning in stochastic games by far better than piecewise replicators.

6. CONCLUSIONS

We identified shortcomings of piecewise replicator dynamics, i.e. discontinuities and occurrences of qualitative anomalies, and ascertained cause and effect. State-coupled replicator dynamics were proposed to alleviate these disadvantages. The preceding formalization of piecewise replicators was deliberately factored into the new approach. Finally, this work delivered a comparative study of finite action-set learning automata as well as piecewise and state-coupled replicator dynamics. State-coupled replicators have been shown to predict learning dynamics in stochastic games more accurately than their predecessor, the piecewise model.

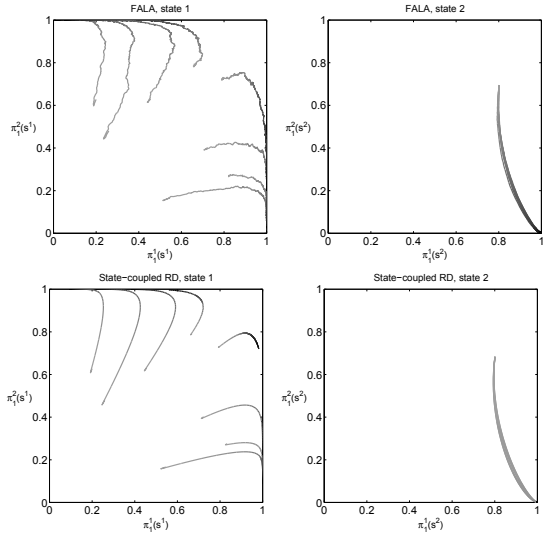

Figure 6: Comparison between trajectory traces of learning automata and state-coupled replicator dynamics in the *Common Interest Game*. Initial action probabilities are fixed in state 2 while a set of 8 random strategy profiles is used in state 1.

This research was partially funded by the Netherlands Organisation for Scientific Research (NWO).

7. REFERENCES

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

[2] D. Hennes, K. Tuyls, and M. Rauterberg. Formalizing multi-state learning dynamics. In *IAT*, 2008.

[3] K. Narendra and M. Thathachar. *Learning Automata An Introduction*. Prentice-Hall, Inc., New Jersey, 1989.

[4] 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.

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

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

[7] K. Tuyls, K. Verbeeck, and T. Lenaerts. A selection-mutation model for Q-learning in multi-agent systems. In *AAMAS*, 2003.

[8] K. Verbeeck, P. Vrancx, and A. Nowé. Networks of learning automata and limiting games. In *ALAMAS*, 2006.

[9] P. Vrancx, K. Tuyls, R. Westra, and A. Nowé. Switching dynamics of multi-agent learning. In *AAMAS*, 2008.