2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

Formalizing Multi-State Learning Dynamics

Daniel Hennes d.hennes@tue.nl
Karl Tuyls k.p.tuyls@tue.nl
Matthias Rauterberg g.w.m.rauterberg@tue.nl

Eindhoven University of Technology, Postbox 513, 5600 MB Eindhoven, The Netherlands
Industrial Design Department, Designed Intelligence Research Group

Abstract

This paper extends the link between evolutionary game theory and multi-agent reinforcement learning to multi-state games. In previous work, we introduced piecewise replicator dynamics, a combination of replicators and piecewise models to account for multi-state problems. We formalize this promising proof of concept and provide definitions for the notion of average reward games, pure equilibrium cells and finally, piecewise replicator dynamics. These definitions are general in the number of agents and states. Results show that piecewise replicator dynamics qualitatively approximate multi-agent reinforcement learning in 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, 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. In [9] we have made the first attempt to extend replicator dynamics to multi-state games. More precisely, we have combined replicator dynamics and piecewise dynamics, called *piecewise replicator dynamics*, to model the learning behavior of agents in stochastic games.

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. Piecewise linear systems are especially suited to be analyzed mathematically due to their reduced complexity. However, they are often still capable of demonstrating the essential qualitative features of many non-linear dynamical systems. Applications include the analysis of regulatory networks e.g. in the area of biology and genomics [2]. Here, we show how this line of reasoning can be translated to replicator dynamics.

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. Section 3 formally introduces piecewise replicator dynamics followed by an example in Section 4. Section 5 covers the experimental results. Section 6 concludes this article.

978-0-7695-3496-1/08 $25.00 © 2008 IEEE
DOI 10.1109/WIIAT.2008.33

2 Background

In this paper, 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 2.1) to model this learning process and use learning automata as an example for reinforcement learning (Section 2.3).

2.1 Replicator dynamics

The continuous 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. A formal definition of an ergodic set 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 $\tau^2(s,a) = B^s_{a_1,a_2}$ under 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 $q_{s^1}\left(s^1,a\right)=Q^{s^1\rightarrow s^1}_{a_1,a_2}=1-Q^{s^1\rightarrow s^2}_{a_1,a_2}$ and $q_{s^2}\left(s^2,a\right)=Q^{s^2\rightarrow s^2}_{a_1,a_2}=1-Q^{s^2\rightarrow s^1}_{a_1,a_2}$.

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.

2.3 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 timeframe 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 timeframe 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 Formalizing 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

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

where

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

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

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

where

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

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 with $G = \langle n, S, A, q, \tau, \pi^1 \dots \pi^n \rangle$ where S itself is the only ergodic set in $S = (s^1 \dots s^k)$, we define the average reward game for some state $s \in S$ as the normal-form game $\bar{G}(s, \pi^1 \dots \pi^n) = \langle n, A^1(s) \dots A^n(s), \bar{\tau}, \pi^1(s) \dots \pi^n(s) \rangle$, 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*

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

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:

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

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 $\pi = \left(\pi^1 \ldots \pi^n\right)$ under which the following condition holds*

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

where $\bar{\tau}$ is the payoff function of the average reward game $\bar{G}\left(s, \pi^1 \ldots \pi^n\right)$; 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 strategy profiles $\pi\left(s^1\right) \ldots \pi\left(s^{i-1}\right), \pi\left(s^{i+1}\right) \ldots \pi\left(s^k\right)$, 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 for the specified state and cell.

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

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

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 Example

We now consider the *2-State Prisoners' Dilemma* as an example. The payoff $\bar{\tau}\left(s^1, a\right)$ for joint action $a = (2,1)$ (i.e. player 1 *defects* and player 2 *cooperates*) in the average reward game $\bar{G}\left(s^1, \pi\right)$ is computed as follows.

Strategies for player 1 and 2 in state s^2 are defined as

$$\pi^1\left(s^2\right) = \begin{pmatrix} p \\ 1-p \end{pmatrix} \text{ and } \pi^2\left(s^2\right) = \begin{pmatrix} q \\ 1-q \end{pmatrix}.$$

The conditions for the stationary distribution $x\left(s^1, a\right)$ simplify from Definition 3 to the following system. For sake of readability, function parameters s^1 and a are omitted for x_{s^1}, x_{s^2}, π^1 and π^2. Corresponding values (see Section 2.2.1) are substituted for Q, A and B.

$$
\begin{aligned}
x_{s^1} + x_{s^2} &= 1 \\
x_{s^1} &= x_{s^2} \pi^{1\prime} Q^{s^2 \to s^1} \pi^2 + x_{s^1} Q_{2,1}^{s^1 \to s^1} \\
x_{s^2} &= x_{s^2} \pi^{1\prime} Q^{s^2 \to s^2} \pi^2 + x_{s^1} Q_{2,1}^{s^1 \to s^2}
\end{aligned}
$$

$$
\begin{aligned}
x_{s^1} &= \frac{0.1 - 0.8\,(2pq - p - q)}{1 - 0.8\,(2pq - p - q)} \\
x_{s^2} &= \frac{0.9 + 0.8\,(2pq - p - q)}{1 - 0.8\,(2pq - p - q)}
\end{aligned}
$$

The average rewards in state s^2 for fixed strategies $\pi\left(s^2\right)$ are as follows. Player 1 receives an average payoff of

$$
\begin{aligned}
P^1 = \sum_{a' \in \prod_{i=1}^{n} A^i(s^2)} \left(\tau^1\left(s^2, a'\right) \prod_{i=1}^{n} \pi_{a_i'}^i\left(s^2\right) \right) &= \pi^{1\prime} A \pi^2 \\
&= 1 - 5pq - p + 9q
\end{aligned}
$$

and player 2 receives $P^2 = 1 - 5pq + 9p - q$.

Thus, player 1's payoff $\bar{\tau}^1\left(s^1, a\right)$ for joint action $a = (2,1)$ in the average reward game $\bar{G}\left(s^1, \pi\right)$ is

$$
\begin{aligned}
\bar{\tau}^1\left(s^1, a\right) = x_{s^2} P^1 + x_{s^1} \tau\left(s^1, a\right) &= x_{s^2} P^1 + x_{s^1} A_{2,1}^1 \\
&= \left(1 - 5pq - p + 9q\right) x_{s^2} + 10 x_{s^1}
\end{aligned}
$$

Similarly, we can compute the average reward game payoffs for player 2 and all other joint actions in state s^1. The immediate payoff $\tau\left(s^1, a\right)$ as well as the stationary distribution x change with a different joint action, while the average reward in s^2 is independent of a.

We can now set up the inequalities relating p and q such that joint action $a = (2,1)$ specifies a pure equilibrium in the average reward game. In this case, Definition 5 translates to the following set of inequality constraints.

$$
\begin{aligned}
\bar{\tau}^1\left(s^1, (2,1)\right) &\geq \bar{\tau}^1\left(s^1, (1,1)\right) \\
\bar{\tau}^2\left(s^1, (2,1)\right) &\geq \bar{\tau}^2\left(s^1, (2,2)\right)
\end{aligned} \tag{5}
$$

If player 1 deviates to action 1 he does not get a higher payoff and the same accounts for player 2 deviating to action 2. After substituting the computed payoffs in (5) we can solve the the system for $p, q \in [0,1]$.

$$
q \leq \frac{7\left(p - \frac{3}{14}\right)}{3 + p} \tag{6}
$$

Figure 1 highlights (thick line) the boundary of the pure equilibrium cell according to (6). The area under the graph corresponds to the strategy space in state s^2 under which $a = (2,1)$ is a pure equilibrium in state s^1.

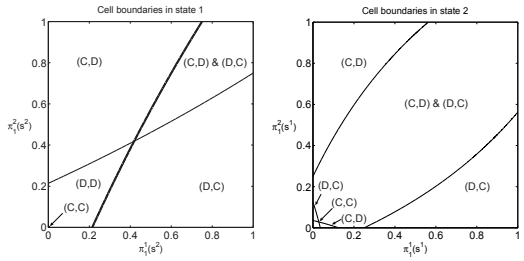

Figure 1: Discrete cell partitioning for the *2-State Prisoners' Dilemma*. Left: Pure equilibria in state 1 as a function of strategies in state 2. Boundary of the pure equilibrium cell for joint action $a = (2,1)$ is highlighted. Right: Pure equilibria in state 2 as a function of strategies in state 1.

5 Results

We consider three distinct types of results to illustrate the strength of our approach. First, the computations of the previous example are extended to determine the full strategy space partitioning for the *2-State Prisoners' Dilemma*. The resulting cell boundaries are plotted in Figure 1. Since the game features only two states, the average reward game in state 1 is completely determined by the strategies in state 2 and vice versa. This means we are able to illustrate the strategy space partitioning in a two dimensional plot. Note that the boundary computed in Section 4 for state 1 is now intersected by another graph. This means, the two cells corresponding to the joint actions *defect-cooperate* and *cooperate-defect* overlap for some subspace of the strategy space in state 2. The partitioning in state 2 shows another interesting property besides cell-overlap. We observe that the strategy space of a cell (this includes cells corresponding to equilibria combinations) might be disjoint. In fact, this is the case for joint actions *defect-cooperate* and *cooperate-defect* in state 2.

Second, we consider a sample trajectory trace for an automata game. Figure 2 shows a sequence of snapshots. Each still image is built up of three layers; the learning trace of automata, cell partitioning and a vector field. The learning trace in state 1 is plotted together with the cell boundaries for state 2 and vice versa. Depending on the current endpoint 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. This layered sequence plot is an excellent tool to convey an intuition on how piecewise replicator dynamics work. Each boundary intersection in state 1 causes a qualitative change of dynamics in state 2 while the vector field layer predicts the movement of the learning trajectory.

Third, 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

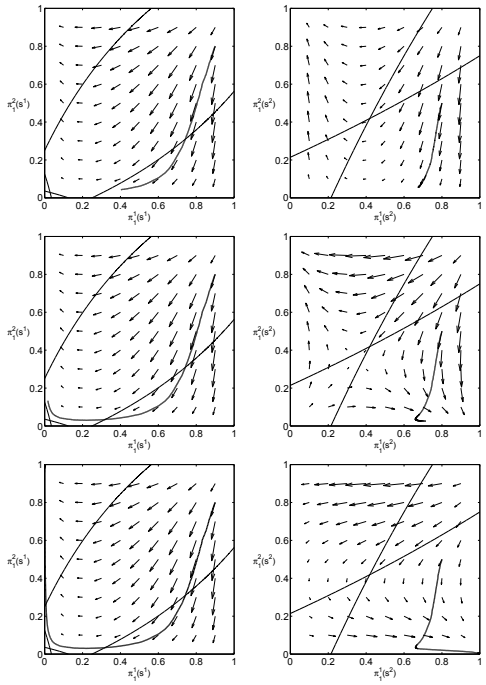

Figure 2: Trajectory plot for learning automata in the *2-State Prisoners' Dilemma*. Each boundary intersection in state 1 causes a qualitative change of dynamics in state 2.

predictive quality of piecewise replicator dynamics with respect to the learning curves of automata games as shown in Figure 3.

6 Discussion and conclusions

The formalization of piecewise replicator dynamics entails considerable advantages. First, this paper has presented a method to theoretically derive cell boundaries which is superior to the empirical approach described in [9]. The theoretical method allows a correct and conclusive analysis, even for extreme point situations. The latter might be fault-prone due to coarse strategy space sampling. In fact, the (C, C) cell in state 1 and the disjoint strategy space for cells (C, D) and (D, C) in state 2 have only been detected analytically. The cell partitioning in Figure 1 therefore refines previous results.

Second, the explicit replicator equations given in Definition 6 now allow to perform trajectory analysis in addition to visualizing basins of attraction. The analysis of trajectory traces gives more insight into the learning dynamics and allows a direct comparison between automata and replicator trajectories. As a matter of fact, our results affirm the proof of concept in previous work [9] and show that piecewise replicator dynamics qualitatively approximate multi-agent reinforcement learning in stochastic games.

To conclude, this paper has further extended the link between evolutionary game theory and multi-agent reinforcement learning to multi-state games. We augmented previous work by general definitions for the concepts of average reward games, pure equilibrium cells and piecewise replicators and illustrated the methodology. Most importantly, this work has provided a formalization of piecewise replicator dynamics, the first approach to model multi-state reinforcement learning dynamics.

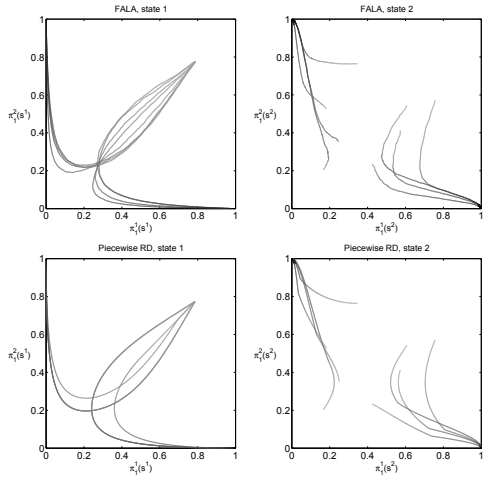

Figure 3: Comparison between trajectory traces of FALA and piecewise replicator dynamics in the *2-State Prisoners' Dilemma*. Initial action probabilities are fixed in state 1 while a set of 8 random strategy profiles is used in state 2.

This research was partially funded by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO).

References

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

[2] H. de Jong, M. Page, C. Hernandez, and J. Geiselmann. Qualitative simulation of genetic regulatory networks: Method and application. In *Proceedings of IJCAI*, 2001.

[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.