Journal of Artificial Intelligence Research 53 (2015) 659-697 Submitted 4/15; published 8/15

Evolutionary Dynamics of Multi-Agent Learning: A Survey

Daan Bloembergen D.BLOEMBERGEN@LIVERPOOL.AC.UK
Karl Tuyls K.TUYLS@LIVERPOOL.AC.UK
Department of Computer Science, University of Liverpool
Ashton Building, Ashton Street, Liverpool L69 3BX, UK

Daniel Hennes DANIEL.HENNES@ESA.INT
Advanced Concepts Team, European Space Agency
Keplerlaan 1, 2201 AZ Noordwijk, NL

Michael Kaisers M.KAISERS@CWI.NL
Centrum Wiskunde & Informatica
Science Park 123, 1098 XG Amsterdam, NL

Abstract

The interaction of multiple autonomous agents gives rise to highly dynamic and nondeterministic environments, contributing to the complexity in applications such as automated financial markets, smart grids, or robotics. Due to the sheer number of situations that may arise, it is not possible to foresee and program the optimal behaviour for all agents beforehand. Consequently, it becomes essential for the success of the system that the agents can *learn* their optimal behaviour and adapt to new situations or circumstances. The past two decades have seen the emergence of reinforcement learning, both in single and multi-agent settings, as a strong, robust and adaptive learning paradigm. Progress has been substantial, and a wide range of algorithms are now available. An important challenge in the domain of multi-agent learning is to gain qualitative insights into the resulting system dynamics. In the past decade, tools and methods from evolutionary game theory have been successfully employed to study multi-agent learning dynamics formally in strategic interactions. This article surveys the dynamical models that have been derived for various multi-agent reinforcement learning algorithms, making it possible to study and compare them qualitatively. Furthermore, new learning algorithms that have been introduced using these evolutionary game theoretic tools are reviewed. The evolutionary models can be used to study complex strategic interactions. Examples of such analysis are given for the domains of automated trading in stock markets and collision avoidance in multi-robot systems. The paper provides a roadmap on the progress that has been achieved in analysing the evolutionary dynamics of multi-agent learning by highlighting the main results and accomplishments.

1. Introduction

In a multi-agent system, several autonomous agents interact in the same environment. Therefore, multi-agent systems can be used to model many complex problems of today's society, such as urban and air traffic control (Agogino & Tumer, 2012), multi-robot coordination (Ahmadi & Stone, 2006; Claes, Hennes, Tuyls, & Meeussen, 2012), distributed sensing (Mihaylov, Tuyls, & Nowé, 2014), energy distribution (Pipattanasomporn, Feroze, & Rahman, 2009), and load balancing (Schaerf, Shoham, & Tennenholtz, 1995; Verbeeck, Nowé, & Tuyls, 2005). The fact that multiple agents interact leads to a highly dynamic, non-deterministic environment. In such an environment, defining proper behaviour for each agent in advance is non-trivial and therefore *learning* is crucial. Recent publications at agents and machine learning conferences, as well as papers published in related mainstream journals, make clear that the number of newly proposed multi-agent learning algorithms is constantly growing. An overview of well-established multi-agent learning algorithms with their various purposes can be attained from previous multi-agent learning survey papers (Panait & Luke, 2005; 't Hoen, Tuyls, Panait, Luke, & Poutré, 2005; Busoniu, Babuska, & De Schutter, 2008; Tuyls & Weiss, 2012), and demonstrates the need for a comprehensive understanding of their qualitative similarities and differences. Although single-agent learning – in particular reinforcement learning – has been extensively studied and acquired a strong theoretical foundation (Kaelbling, Littman, & Moore, 1996; Sutton & Barto, 1998), a thorough understanding of learning in *multi-agent* settings has long remained an open problem (Tuyls, 't Hoen, & Vanschoenwinkel, 2006).

Learning in multi-agent systems is not only relevant within the field of artificial intelligence but has been extensively studied in game theory and economics as well (Shoham, Powers, & Grenager, 2007). It is not surprising then, that these fields share a lot of common ground. Indeed, game theory often provides the context in which multi-agent systems are modelled and evaluated. Recently, some multi-agent learning research has shifted its focus from traditional game theory to *evolutionary game theory* (Tuyls et al., 2006; Tuyls & Parsons, 2007). The concepts employed by evolutionary game theory prove well suited to describe learning in multi-agent systems. Both fields are concerned with dynamic environments with a high level of uncertainty, characterised by the fact that agents lack complete information (Tuyls et al., 2006). Moreover, there exists a formal relation between the behaviour of one of the most basic reinforcement learning algorithms, Cross learning (Cross, 1973), and the population dynamics of evolutionary game theory, described by the replicator dynamics (Börgers & Sarin, 1997). Although this link was originally established in the context of stateless normal-form games only, it has since been extended to more complex scenarios as well (e.g., Hennes, Tuyls, & Rauterberg, 2009; Tuyls & Westra, 2009; Galstyan, 2013; Panozzo, Gatti, & Restelli, 2014).

Understanding this relation sheds light into the black box of reinforcement learning by making it possible to analyse the learning dynamics of multi-agent systems in detail and to compare the behaviour of different algorithms in a principled manner. This in turn facilitates important tasks such as parameter tuning, and helps in selecting a specific learner for a given problem. Tuyls and Nowé (2005) and Tuyls et al. (2006) were the first to present an overview of this evolutionary game theoretic approach to multi-agent learning. They build on the connection between Cross learning and the replicator dynamics and extend this link to learning automata and Q-learning as well. However, much progress has been made in the past decade, warranting an up-to-date overview and roadmap of this research area. This is precisely our aim in this work.

We believe that evolutionary game theory has a lot to offer for both the understanding and application of multi-agent learning. Shoham et al. (2007) call for a more grounded approach to research in multi-agent learning, suggesting five agendas to which such research could contribute. They also caution not to rely too strongly on requirements such as convergence to a Nash equilibrium when evaluating learning algorithms in a multi-agent setting. In response, Tuyls and Parsons (2007) argue in favour of evolutionary game theory, rather than classical game theory, as the preferable framework within which to study multi-agent learning formally. They show how research on the evolutionary framework contributes to each of the five agendas of research identified by Shoham et al. (2007). Moreover, it allows us to move away from the static Nash equilibrium, and focus instead on the transient dynamics of the learning process.

In this article we describe the formal relation between evolutionary game theory and multi-agent learning in detail and survey the recent advances and extensions that have been made in this area.[1] To this end we present a categorisation of related work based on the nature of the environment (stateless or multi-state games) and the actions (discrete or continuous) available to the learning agents. Moreover, we provide examples of the successful application of this approach in relation to parameter tuning, the design of new learning algorithms, and the analysis of complex strategic interactions such as automated trading and multi-robot collision avoidance. The evolutionary game theoretic approach offers a promising new paradigm within which to study multi-agent learning, as it provides new insights towards the understanding, analysis, and design of multi-agent reinforcement learning algorithms.

The paper proceeds as follows. Section 2 provides the necessary background on (multi-agent) reinforcement learning and evolutionary game theory. Section 3 introduces the link between the replicator dynamics and Cross learning and outlines the methodology of this survey. An overview of recent advances and extensions is given in Section 4, supported by empirical validation in Section 5. Section 6 presents examples of the application of the evolutionary game theoretic approach. Section 7 concludes the article.

2. Preliminaries

In this section, we outline the fundamentals of multi-agent learning. Firstly, we concisely present Markov decision processes (MDPs) as the standard formal framework for single-agent decision making, together with a well-known reinforcement learning algorithm, Q-learning. We then proceed to present stochastic games, also called Markov games, as a multi-agent extension to MDPs, as well as three approaches to learning in this extended setting: independent learning, joint action learning, and gradient based methods. Finally, we discuss how evolutionary game theory can be used to reason about multi-agent interactions, paving the way to formally relate these two fields in Section 3.

2.1 Reinforcement Learning

Reinforcement learning is based on the concept of trial-and-error learning, which underlies many theories of (human) learning and intelligence (Sutton & Barto, 1998). The reinforcement learning agent continuously interacts with the environment, perceiving its state, taking actions, and observing the effect of those actions (see Figure 1). Actions that yield a positive effect will have a higher chance of being executed again in the future, that is to

1. We do not attempt to present a broad survey on multi-agent learning in general but focus solely on those works that explicitly model multi-agent learning using methods from evolutionary game theory. An excellent survey and taxonomy of multi-agent learning algorithms in general can be found in the work of Busoniu et al. (2008).

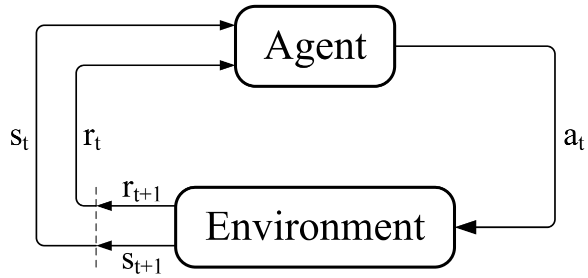


Figure 1: A reinforcement learning agent perceives state s_t of the environment at time t, decides to take action a_t, upon which the environment transitions to state s_{t+1} and the agent receives reward r_{t+1}.

say they are *reinforced* within the agent's behaviour. To this effect, the agent receives a reward signal that indicates the quality of the actions taken; however, the reward may be stochastic, delayed, or accumulated over sequences of actions. Therefore, the agent needs to balance exploration and exploitation, to avoid getting stuck in local optima. The objective of the learning agent is to discover a policy, represented as a mapping from states to actions, that maximises its long-term expected reward.

The single-agent reinforcement learning setting can be formalised as a *Markov decision process* (MDP) (Puterman, 1994). An MDP is defined by finite state and action sets, S and A, one-step state transition dynamics

$$\mathcal{P}^a_{ss'} = P(s_{t+1} = s' | s_t = s, a_t = a) \tag{1}$$

describing the probability of transitioning to state $s' \in S$ after taking action $a \in A$ in state s, and the expected value of the next reward

$$\mathcal{R}^a_{ss'} = E(r_{t+1} | s_t = s, a_t = a, s_{t+1} = s') \tag{2}$$

given the previously executed action and resulting state transition. Both state transitions and rewards can be stochastic, and in fact learning these stochastic models is a key task in many reinforcement learning problems. The learning goal in an MDP is to find a policy π that maps states to action selection probabilities, maximising expected reward. When following a fixed policy π we can define the value of a state s under that policy as the total amount of reward R the agent expects to accumulate when starting in state s and following π thereafter:

$$V^\pi(s) = E_\pi(R_t | s_t = s) = E_\pi(\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s).$$

The rewards are discounted by factor $\gamma \in [0, 1)$ to ensure a bounded sum in infinite horizon MDPs. The value function for a policy π can be calculated iteratively using the *Bellman equation* (Bellman, 1957). Starting with an arbitrarily chosen value function V^π, at each iteration and for each state s the value function is then updated based on the immediate reward and the current estimate of V^π:

$$V^\pi(s) \leftarrow \sum_{a \in A(s)} \pi(s, a) \sum_{s' \in S} \mathcal{P}^a_{ss'} [\mathcal{R}^a_{ss'} + \gamma V^\pi(s')].$$

The Bellman equation expresses the recursive relation between the value of a state and its successor states, and averages over all possibilities, weighing each by its probability of occurring. In this setting, finding an optimal policy π^* is equivalent to finding a policy that maximises the value function, i.e.,

$$V^{\pi^*}(s) = \max_{\pi} V^{\pi}(s) \quad \forall s \in S.$$

When a model of the environment is available, in particular if $\mathcal{P}$ and $\mathcal{R}$ are known, the Bellman equation can be applied to compute an optimal policy directly, using a dynamic programming technique such as value iteration or policy iteration (Sutton & Barto, 1998). In general, however, such a model may not be available. In this case, reinforcement learning can be used to find an optimal mapping from states to actions. Arguably the most famous example of a reinforcement learning algorithm is the model-free temporal difference algorithm *Q-learning* (Watkins & Dayan, 1992). Q-learning maintains a value function over state-action pairs, $Q(s, a)$, which it updates based on the immediate reward and the discounted expected future reward according to Q:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a} Q(s_{t+1}, a) - Q(s_t, a_t) \right]. \tag{3}$$

Here, γ is the discount factor for future rewards as before, and $\alpha \in [0, 1]$ is the learning rate that determines how quickly Q is updated based on new reward information. Q-learning is proven to converge to the optimal policy, given "sufficient" updates for each state-action pair, and a decreasing learning rate $\alpha \to 0$ (Watkins & Dayan, 1992).

Choosing which action to take is a crucial aspect of the learning process. Should the agent exploit actions that yielded high reward in the past, or should it explore in order to achieve potentially better results in the future, thereby risking a low reward now? Neither of the two is sufficient on its own, and the dilemma is to find the right balance (Kaelbling et al., 1996; Sutton & Barto, 1998). Two often used action selection mechanisms are *ϵ-greedy* and *softmax* or *Boltzmann* exploration (Sutton & Barto, 1998). ϵ-Greedy selects the best action (greedy w.r.t. Q) with probability $1 - \epsilon$, and with probability ϵ it selects an action at random. The Boltzmann exploration mechanism makes use of a temperature parameter τ that controls the balance between exploration and exploitation. Action a_i is chosen in state s with probability

$$p_i = \frac{e^{Q(s,a_i)/\tau}}{\sum_j e^{Q(s,a_j)/\tau}}. \tag{4}$$

A high temperature drives the mechanism towards exploration, whereas a low temperature promotes exploitation, favouring actions with higher Q-values.

2.2 From Single-Agent to Multi-Agent Learning

The MDP framework assumes that a single agent is active in the environment. Once multiple agents interact and learn simultaneously, the model needs to be extended. *Stochastic games*, or Markov games, offer a generalisation of MDPs to the multi-agent domain (Littman, 1994). In a stochastic game, each agent has its own set of actions, i.e., for n agents the joint-action space is $A = A^1 \times A^2 \times \ldots \times A^n$. The state transition and reward functions now depend on the joint action of all agents:

$$\mathcal{R} : S \times A^1 \times \ldots \times A^n \times S \mapsto \mathbb{R}^n$$

$$\mathcal{P} : S \times A^1 \times \ldots \times A^n \times S \mapsto [0, 1].$$

The immediate rewards may be the same for all agents but they need not be in general. A special case of stochastic games is the stateless setting described by *normal-form games*. Normal-form games are one-shot interactions, where all agents simultaneously select an action and receive a reward based on their joint action, after which the game ends. There is no state transition function, and the reward function can be represented by an n-dimensional payoff matrix, for n agents. An agent's policy is simply a probability distribution over its actions. Repeated normal form games are common benchmarks for multi-agent learning. Such scenarios are detailed further in Section 2.3.

Learning in a multi-agent setting is inherently more complex than in the single-agent case described previously, as agents interact both with the environment and potentially with each other. Learning is simultaneous, meaning that changes in the policy of one agent may affect the rewards and hence the optimal policy of others. Moreover, agents may have conflicting interests, yet cooperation with competitors may yield short or long term benefits. This makes it difficult to judge the learning process, since myopic maximisation of individual rewards might not lead to the best overall solution. The fact that the reward function depends on the actions of other agents leads to an important characteristic of multi-agent reinforcement learning: the environment is *non-stationary* and as a result each agent is essentially pursuing a moving target (Busoniu et al., 2008). Moreover, the fact that multiple agents influence the environment means that, from the perspective of the individual agents, the Markov property no longer holds. Two different approaches to multi-agent learning can be distinguished: independent learning and joint-action learning (Claus & Boutilier, 1998). In the following we briefly discuss both approaches, and list notable algorithms in each class. For a detailed taxonomy of multi-agent learning algorithms, we refer to the excellent survey of Busoniu et al. (2008).

2.2.1 Independent Learning

Independent learners mutually ignore each other, thereby effectively reducing the multi-agent learning problem to a single-agent one. Interaction with other agents is implicitly perceived as noise in a stochastic environment. The advantage of this approach is that single-agent learning algorithms can straightforwardly be applied to a multi-agent setting, and scalability in the number of agents is not an issue.[2] However, stochasticity of the environment means that convergence guarantees from the single-agent setting are lost. In particular, the Markov property on which such proofs are typically based, no longer holds. Moreover, no explicit mechanism for coordination is available to the agents. Despite these drawbacks, independent learners have shown good performance in many multi-agent settings (Busoniu et al., 2008).

2. Computational complexity increases only linearly with the number of agents. Performance may vary depending on the specific domain and algorithm.

Traditional single-agent reinforcement learning algorithms, such as Q-learning (Watkins & Dayan, 1992) and (networks of) learning automata (Narendra & Thathachar, 1974; Wheeler Jr & Narendra, 1986; Vrancx, Verbeeck, & Nowé, 2008b), can be directly applied in this setting. Moreover, various new independent learning algorithms have been proposed specifically with the multi-agent setting in mind. For example, Bowling and Veloso (2001) proposed *policy hill climbing* with the *win or learn fast* heuristic (WoLF-PHC), and show that the algorithm is rational and convergent in multi-agent domains. Other examples include *frequency maximum Q-learning* (Kapetanakis & Kudenko, 2002), an algorithm tailored to coordinate in cooperative multi-agent systems, and a class of *regret minimisation* algorithms (Blum & Mansour, 2007) that guarantee performance close to the best fixed action in hindsight against any opponent. Finally, two extensions to Q-learning have been proposed that alleviate certain artifacts of this algorithm in non-stationary (e.g. multi-agent) environments: *frequency-adjusted Q-learning* (Kaisers & Tuyls, 2010), and *repeated update Q-learning* (Abdallah & Kaisers, 2013). Frequency-adjusted Q-learning has been proven to converge in two-player two-action normal-form games (Kaisers & Tuyls, 2011; Kianercy & Galstyan, 2012).

2.2.2 Joint-Action Learning

Whereas independent learners completely ignore the presence of other agents, *joint-action learners* explicitly take them into account. Joint-action learners achieve this by learning in the space of joint actions, rather than in their individual action space only (Claus & Boutilier, 1998). They observe the actions of other agents in order to estimate their policy, and then act optimally given those estimated policies. This way, joint action learners have better means of coordination. The drawback is that the agent needs to be able to observe the other agents' actions, and assumptions about the opponents' adaptation mechanism are necessary to derive reasonable predictions of the opponents' future actions. Moreover, the complexity of the algorithm grows exponentially with the number of agents. Examples of joint action learners are *minimax-Q* (Littman, 1994), *fictitious play* and AWESOME (Brown, 1951; Conitzer & Sandholm, 2007), *hyper-Q* (Tesauro, 2003), and *Nash-Q* (Hu & Wellman, 2003). Worth mentioning here as well is the related stream of work on *Bayesian reinforcement learning* (Dearden, Friedman, & Russell, 1998; Strens, 2000). Of particular interest to the discussion of multi-agent learning is the work of Chalkiadakis and Boutilier (2003), who use the Bayesian framework to explicitly model an agent's uncertainty about both the model of the environment and the strategies of the other agents.

2.2.3 Gradient Ascent Optimisation

A somewhat separate stream of multi-agent learning research revolves around *gradient ascent* based algorithms. These methods often fall in between independent learning and joint-action learning, but are worth mentioning separately as they are important for our discussion in Section 4.1. Gradient ascent (or descent) is a well-known optimisation technique in the field of machine learning. Given a well-defined differentiable objective function, the learning process can follow the direction of its gradient in order to find a local optimum. This concept can be adapted for multi-agent learning by having the learning agents' policies follow the gradient of their individual expected reward.

Examples of gradient ascent algorithms are *infinitesimal gradient ascent* (IGA), which is designed specifically for two-player two-action normal-form games (Singh, Kearns, & Mansour, 2000), and *generalized infinitesimal gradient ascent* (GIGA), which extends IGA to games with an arbitrary number of actions (Zinkevich, 2003). Both algorithms can be combined with the *win or learn fast* (WoLF) heuristic in order to improve convergence in stochastic games (Bowling & Veloso, 2002; Bowling, 2005). Both IGA and GIGA assume that the agents have knowledge of the (reward) structure of the game, or at least have some mechanism for approximating the gradient of the value function, which is not generally feasible in practice. However, the more recent algorithm *weighted policy learning* (WPL) relaxes this assumption (Abdallah & Lesser, 2008).

2.3 Game Theory

Game theory (Von Neumann & Morgenstern, 1944; Gibbons, 1992) is a theory of interactive strategic decision making, and is therefore of utmost importance for multi-agent systems. It studies this decision making in the form of cooperative and competitive games. In these games, each player has a set of actions and a preference over the joint action outcome, which is captured by a numerical payoff signal. For games between two players that are played only once, i.e., one-shot two-player games, the payoffs can be represented by a bi-matrix $(\mathbf{A}, \mathbf{B})$, which gives the payoff for the row player in $\mathbf{A}$, and the column player in $\mathbf{B}$ (see Figure 2). In this example, the row player chooses one of the two rows, the column player chooses on of the columns, and the outcome of their joint action determines the payoff to both. The goal for each player is to come up with a strategy (a probability distribution over his actions) that maximises their expected payoff in the game. Note that games, players, strategies, and payoffs of game theory map one-to-one to environments, agents, policies, and rewards in the multi-agent systems literature.

The players are thought of as individually rational, in the sense that each player is perfectly logical and tries to maximise his own payoff, assuming the others are doing likewise. Under this assumption, the *Nash equilibrium* (NE) solution concept can be used to study what players will reasonably choose to do. A set of strategies forms a NE if no single player can do better by unilaterally switching to a different strategy. In other words, each strategy in a NE is a best response against all other strategies in that equilibrium.

A game can have more than one Nash equilibrium, some of which may not be preferred equally. Moreover, the NE may not be the best outcome from a social point of view. As an example, consider the Prisoner's Dilemma (Axelrod & Hamilton, 1981), depicted in Figure 3 (left). In this game, two players simultaneously choose either to cooperate (C) or defect (D). Individually, defection is a best response against any opponent strategy, and as a result mutual defection is the single Nash equilibrium of the game. However, both players would be better off if both would cooperate – hence the dilemma.

$$\begin{pmatrix} a_{11}, b_{11} & a_{12}, b_{12} \\ a_{21}, b_{21} & a_{22}, b_{22} \end{pmatrix}$$

Figure 2: General payoff bi-matrix $(\mathbf{A}, \mathbf{B})$ for a two-player two-action normal form game.

$$\begin{array}{cc} & \begin{array}{cc} C & D \end{array} \\ \begin{array}{c} C \\ D \end{array} & \begin{pmatrix} 3,3 & 0,5 \\ 5,0 & 1,1 \end{pmatrix} \end{array} \qquad \begin{array}{cc} & \begin{array}{cc} S & H \end{array} \\ \begin{array}{c} S \\ H \end{array} & \begin{pmatrix} 4,4 & 1,3 \\ 3,1 & 3,3 \end{pmatrix} \end{array} \qquad \begin{array}{cc} & \begin{array}{cc} H & T \end{array} \\ \begin{array}{c} H \\ T \end{array} & \begin{pmatrix} 1,0 & 0,1 \\ 0,1 & 1,0 \end{pmatrix} \end{array}$$

Figure 3: Payoff matrix for the Prisoner's Dilemma (left), the Stag Hunt (center), and Matching Pennies (right).

A second example is given by the Stag Hunt (Skyrms, 2004), shown in Figure 3 (center). In this coordination game, both players prefer to jointly choose either to hunt for stag (S) or hare (H). Hunting for hare provides a safe choice, as the payoff for this action is independent of the choice of the opponent. Hunting stag is more risky, however both players yield a higher payoff if they manage to coordinate. This game has two pure Nash equilibria, (S, S) and (H, H), and one mixed Nash equilibrium where both players randomise and play S with probability $\frac{2}{3}$.

Finally, in the Matching Pennies game (Figure 3, right) two players simultaneously choose which side of their coin to display, either heads (H) or tails (T). If both choose the same side, the first player gets to keep both coins. If they pick opposite sides, the second player keeps the coins. In this zero-sum game, the single mixed NE is for both players to randomise uniformly over their actions.

2.4 Evolutionary Game Theory

Classical game theory assumes that full knowledge of the game is available to all players, which together with the assumption of individual rationality does not necessarily reflect the dynamic nature of real world interactions. *Evolutionary game theory* relaxes the rationality assumption and replaces it by biological concepts such as natural selection and mutation (Maynard Smith & Price, 1973; Weibull, 1997; Hofbauer & Sigmund, 1998; Gintis, 2009). Central to evolutionary game theory are the *replicator dynamics* that describe how a population of individuals evolves over time under evolutionary pressure. Each individual is of a certain type, and individuals are randomly paired in interaction. Their reproductive success is determined by their fitness, which results from these interactions. The replicator dynamics dictate that the population share of a certain type will increase if the individuals of this type have a higher fitness than the population average; otherwise their population share will decrease. The population can be described by the state vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^\top$, with $0 \leq x_i \leq 1\ \forall i$ and $\sum_i x_i = 1$, representing the fractions of the population belonging to each of n types. Now suppose the fitness of type i is given by the fitness function $f_i(\mathbf{x})$, and the average fitness of the population is given by $\bar{f}(\mathbf{x}) = \sum_j x_j f_j(\mathbf{x})$. Using $\dot{x}_i$ to denote $\frac{dx_i}{dt}$, the population change over time can then be written as

$$\dot{x}_i = x_i \left[f_i(\mathbf{x}) - \bar{f}(\mathbf{x}) \right]. \tag{5}$$

These replicator dynamics describe the change over time of a large population of individuals. However, the model can be interpreted alternatively as representing the strategy of a single player, where the population share of each type represents the probability with which the player selects the corresponding pure action, as summarised in Table 1. The replicator dynamics now describe the player's strategy change over time as he repeatedly plays the

Table 1: Correspondence of terminology between the domains of reinforcement learning, game theory, and evolutionary game theory.

Reinforcement Learning	Game Theory	Evolutionary Game Theory
environment	game	game
agent	player	population
action	action	type
policy	strategy	distribution over types
reward	payoff	fitness

game and iteratively updates his policy. Evolutionary game theory refines the static Nash equilibrium (NE) concept with the notion of *evolutionarily stable strategies* (ESS). A strategy $\mathbf{x}$ is an ESS if it is immune to invasion by mutant strategies, given that the mutants initially occupy only a small fraction of the population. Let $f(\mathbf{x}, \mathbf{y})$ be the (expected) fitness of strategy $\mathbf{x}$ against strategy $\mathbf{y}$. Formally then, strategy $\mathbf{x}$ is an ESS iff, for any mutant strategy $\mathbf{y}$, the following hold:

1. $f(\mathbf{x}, \mathbf{x}) \geq f(\mathbf{y}, \mathbf{x})$, and
2. if $f(\mathbf{x}, \mathbf{x}) = f(\mathbf{y}, \mathbf{x})$, then $f(\mathbf{x}, \mathbf{y}) > f(\mathbf{y}, \mathbf{y})$.

The first condition states that an ESS is also a NE of the original game. The second condition states that if the invading strategy does as well against the original strategy as the original strategy does against itself, then the original strategy must do better against the invader than the invader does against itself. This means that ESS are a refinement of the NE solution concept. Moreover, every ESS is an asymptotically stable fixed point of the replicator dynamics (Weibull, 1997).

In a two-player game, each player is described by his own evolving population, and at every iteration of the game one individual of each player's population is drawn to interact. Therefore, the fitness of each type now depends on the population distribution of the co-player, i.e., the two populations are co-evolving. If the two players' populations are given by $\mathbf{x}$ and $\mathbf{y}$ and their fitness functions by the payoff matrices $\mathbf{A}$ and $\mathbf{B}$, we can write the expected fitness of type i of population $\mathbf{x}$ as

$$f_i(\mathbf{x}) = \sum_j a_{ij} y_j = (\mathbf{A}\mathbf{y})_i$$

and similarly we can write the average population fitness as

$$\bar{f}(\mathbf{x}) = \sum_i x_i \sum_j a_{ij} y_j = \mathbf{x}^\top \mathbf{A}\mathbf{y}.$$

Following similar reasoning for the population $\mathbf{y}$, we can rewrite Equation 5 for the two populations as

$$\begin{aligned} \dot{x}_i &= x_i \left[(\mathbf{A}\mathbf{y})_i - \mathbf{x}^\top \mathbf{A}\mathbf{y} \right] \\ \dot{y}_i &= y_i \left[(\mathbf{x}^\top \mathbf{B})_i - \mathbf{x}^\top \mathbf{B}\mathbf{y} \right]. \end{aligned} \tag{6}$$

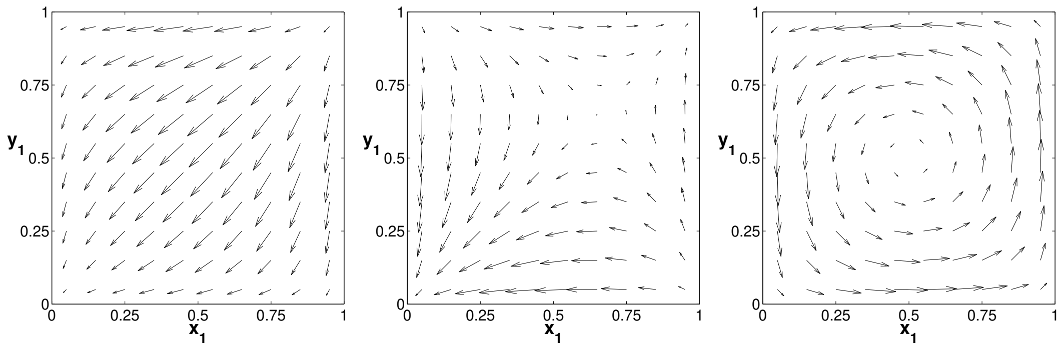


Figure 4: The replicator dynamics, plotted in the unit simplex, for the prisoner's dilemma (left), the stag hunt (center), and matching pennies (right).

To illustrate the dynamics of Equation 6, we analyse the three games presented in Figure 3. Since a player's strategy over two actions is fully defined by the probability of the first action (as $x_2 = 1 - x_1$), we can plot the strategy space of these games as the two-dimensional unit simplex over the tuple (x_1, y_1). Plugging the payoff matrix of each game into the replicator dynamics of Equation 6, we find the direction and relative speed of change for each point in the unit simplex. The resulting vector fields for the three games are shown in Figure 4.

Figure 4 shows that the players in the prisoner's dilemma are drawn to the (D, D) equilibrium, which is both a NE and an ESS. In the stag hunt, both pure NE, (S, S) and (H, H), are also ESS, but the mixed NE is not. It is a fixed point, but not asymptotically stable. Finally, the matching pennies game has a single mixed NE at $(\frac{1}{2}, \frac{1}{2})$, where both players randomise uniformly over their actions. However, this again is not an ESS; instead all trajectories cycle around this fixed point.

3. Relating Reinforcement Learning and Replicator Dynamics

Recent research analysing the dynamics of multi-agent learning builds on seminal work by Börgers and Sarin (1997), who first proved the formal relation between the replicator dynamics of evolutionary game theory and reinforcement learning. In this section, we will first summarise their proof. Next, we present a categorisation of recent work, based on the nature of the environment and actions available to the agents.

3.1 Replicator Dynamics as the Continuous Time Limit of Cross Learning

Multi-agent learning and evolutionary game theory share a substantial part of their foundation, in that they both deal with the decision making processes of boundedly rational agents, or players, in uncertain environments. The link between these two fields is not only an intuitive one, but was made formal with the proof that the continuous time limit of *Cross learning* converges to the replicator dynamics (Börgers & Sarin, 1997).

Cross learning (Cross, 1973) is one of the most basic stateless reinforcement learning algorithms, which updates policy[3] π based on the reward r received after taking action j as

$$\pi(i) \leftarrow \pi(i) + \begin{cases} r - \pi(i)r & \text{if } i = j \\ -\pi(i)r & \text{otherwise} \end{cases}. \tag{7}$$

A valid policy is ensured by the update rule as long as the rewards are normalised, i.e., $0 \leq r \leq 1$. Cross learning is closely related to finite action-set learning automata (Narendra & Thathachar, 1974; Thathachar & Sastry, 2002). In particular, it is equivalent to a learning automaton with a linear reward-inaction (L_{R-I}) update scheme and a learning step size (α) of 1.

We can estimate the expected change in the policy, $E\left[\Delta\pi(i)\right]$, induced by Equation 7 (Börgers & Sarin, 1997). The probability $\pi(i)$ of action i is affected both if i is selected and if another action j is selected. Let $E_i[r]$ be the expected reward after taking action i. We can now write

$$\begin{aligned} E\left[\Delta\pi(i)\right] &= \pi(i)\Big[E_i[r] - \pi(i)E_i[r]\Big] + \sum_{j \neq i} \pi(j)\Big[-E_j[r]\pi(i)\Big] \\ &= \pi(i)\left[E_i[r] - \textstyle\sum_j \pi(j)E_j[r]\right]. \end{aligned} \tag{8}$$

Assuming the learner takes infinitesimally small update steps, the continuous time limit of Equation 8 can be taken as

$$\pi_{t+\delta}(i) = \pi_t(i) + \delta\Delta\pi_t(i)$$

with $\lim \delta \to 0$. This yields a continuous time system which can be expressed with the partial differential equation

$$\dot{\pi}(i) = \pi(i)\left[\mathrm{E}_i[r] - \textstyle\sum_j \pi(j)\mathrm{E}_j[r]\right]$$

In a two-player normal form game, we can write the policy of an agent simply as a probability distribution over actions, i.e. $\pi \equiv \mathbf{x}$. As such defined, and given payoff matrices $\mathbf{A}$ and $\mathbf{B}$ and policies $\mathbf{x}$ and $\mathbf{y}$ for the two players, respectively, this yields

$$\begin{aligned} \dot{x}_i &= x_i\left[(\mathbf{A}\mathbf{y})_i - \mathbf{x}^\top\mathbf{A}\mathbf{y}\right] \\ \dot{y}_i &= y_i\left[(\mathbf{x}^\top\mathbf{B})_i - \mathbf{x}^\top\mathbf{B}\mathbf{y}\right] \end{aligned} \tag{9}$$

which are exactly the multi-population replicator dynamics of Equation 6.

This link can be made explicit not only theoretically but also empirically, as shown in Figure 5. Here, we simulate the learning process of two Cross learners when taking very small policy update steps (by multiplying the update term of Equation 7 by $\alpha = 0.001$), starting at different initial policies, and overlaying the resulting policy traces on the replicator dynamics of Figure 4. As can be observed, the learning traces follow the replicator dynamics precisely. In a similar fashion, dynamical models of different (and more complex) reinforcement learning algorithms can be derived. These are discussed in the following sections.

3. The dependency of the policy on state s is dropped for stateless environments, and the dependence on time is implied but omitted for notational convenience.

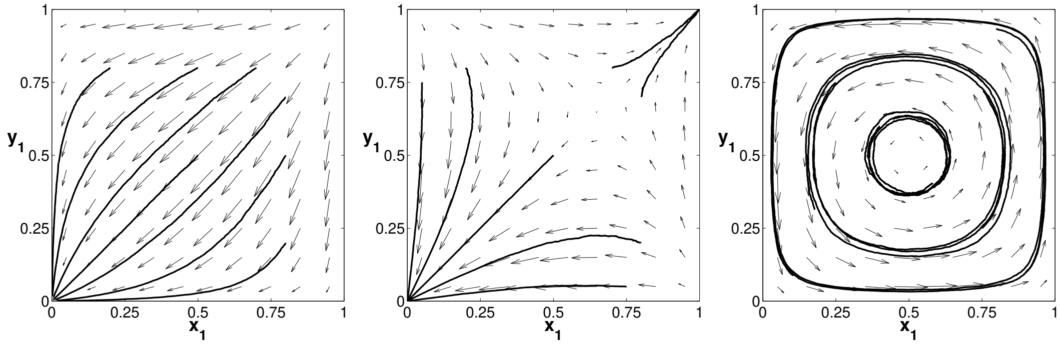


Figure 5: Policy traces of Cross learning, plotted in the unit simplex and overlaid on the replicator dynamics, for the prisoner's dilemma (left), the stag hunt (center), and matching pennies (right).

3.2 Categorisation of Learning Dynamics

We divide the learning algorithms and corresponding dynamics that are presented in this work into four categories, based on the nature of the environment and the actions available to the agent. We distinguish stateless normal-form games, and games with multiple states with probabilistic transitions between them, represented by stochastic games (see also Section 2.2). Moreover, we differentiate between settings where the agent has a finite, discrete choice of actions, and settings which offer a continuous range of choices. Table 2 lists the four resulting categories, along with references to work that has been done in each category. We focus solely on work that explicitly relates the dynamical models to learning in multi-agent systems. A large body of work is available that discusses extensions to the replicator dynamics from an evolutionary game theoretic viewpoint only, however, these fall outside the scope of this survey.[4]

Cross learning, detailed previously, belongs to the first category of stateless games with discrete actions. Other examples in this category are stateless Q-learning and the related frequency adjusted (FAQ) and lenient (LFAQ) versions, regret minimisation, and gradient ascent algorithms. The second category comprises stateless games with a continuous action space. Typically, function approximators are used in such settings (a recent overview is provided in Busoniu, Babuska, De Schutter, & Ernst, 2010), however most work in that category has so far been limited to single-agent learning. Here, we summarise approaches to model such games using continuous action replicator dynamics. The third category is that of stochastic (i.e. multi-state) games with discrete actions. Dynamics have been derived for networks of learning automata, in particular piece-wise and state-coupled replicator dynamics, and the variation RESQ-learning that incorporates exploration in the learning process.

The fourth category in Table 2, comprising stochastic games with continuous action-spaces, is strikingly empty. Indeed, this domain is of main interest for future work, as no attempts have been made so far to derive learning dynamics for this setting. Combining techniques and approaches from the second and third category could be a fruitful starting

4. See e.g. the textbooks by Weibull (1997) and Hofbauer and Sigmund (1998) for an introduction.

Table 2: Categorisation of dynamical models of multi-agent learning that are available in the literature.

	Discrete actions	**Continuous actions**
Normal form games	Q-learning[1] FAQ-learning[2] Regret Minimisation[3] Lenient FAQ-learning[4] Gradient ascent[5]	Continuous action replicator dynamics[6] Q-learning[7]
Stochastic games	Piecewise replicator dynamics[8] State-coupled replicator dynamics[9] RESQ-learning[10]	

[1]Tuyls et al. (2003)
Kianercy and Galstyan (2012)
[2]Kaisers and Tuyls (2010, 2011)
[3]Klos et al. (2010)
[4]Panait et al. (2008)
Bloembergen et al. (2011)
[5]Kaisers et al. (2012)
[6]Tuyls and Westra (2009)
[7]Galstyan (2013)
[8]Vrancx et al. (2008a)
[9]Hennes et al. (2009)
[10]Hennes et al. (2010)

point for such an endeavour. In the next section we provide an overview of the work listed in Table 2, following the same categorisation.

4. Overview of Learning Dynamics

With the categorisation presented in Table 2 in hand, we now give an overview of the dynamics of various multi-agent reinforcement learning algorithms. First, learning dynamics in normal-form games will be discussed. Next, we present replicator dynamics for continuous strategy spaces. Finally, multi-state learning dynamics are described.

4.1 Learning Dynamics in Normal-Form Games

Repeated normal-form games are characterised by being stateless games, in which agents choose from a discrete and finite set of actions at each time step. This greatly simplifies analytical approaches, while at the same time still allowing to capture interesting strategic interactions. As a result, normal-form games have been frequently used as a test-bed for multi-agent learning (Busoniu et al., 2008). Several learning algorithms have been devised specifically for normal-form games; other multi-state algorithms such as Q-learning can straightforwardly be applied by removing the state dependency from the learning update rule. As before, in the remainder we define $\mathbf{x} \equiv \pi$ and $\mathbf{y} \equiv \sigma$ to be the policies of the two agents in the stateless setting.

4.1.1 Independent Reinforcement Learners

As described in Section 3, *Cross learning* (CL) was the first algorithm to be linked to the replicator dynamics of evolutionary game theory (Börgers & Sarin, 1997). In particular, the infinitesimal time limit of the Cross learning update rule (Equation 7) converges to the replicator dynamics. The link between a simple policy learner such as Cross learning, and a dynamical system in the policy space may seem intuitive. However, this link has been extended to value-based (and more complex policy-based) learners as well. A selection-mutation model of Boltzmann Q-learning has been proposed by Tuyls et al. (2003), assuming a constant temperature τ.[5] Tuyls et al. show that the dynamical system can be decomposed into terms for exploitation (selection following the replicator dynamics) and exploration (mutation through randomisation based on the Boltzmann mechanism):[6]

$$\dot{x}_i = \frac{\alpha x_i}{\tau} \underbrace{\left[(\mathbf{Ay})_i - \mathbf{x}^\top \mathbf{Ay}\right]}_{\text{exploitation}} - \alpha x_i \underbrace{\left[\log x_i - \textstyle\sum_k x_k \log x_k\right]}_{\text{exploration}}. \tag{10}$$

Another way to view the two terms of Equation 10 is in relation to the thermodynamical concepts of energy and entropy, where selection is analogous to energy, and mutation to entropy. The entropy term can be further subdivided in the entropy of one individual strategy, $\log x_i$, and the entropy of the entire population, $\sum_k x_k \log x_k$. In this sense, mutation is determined by the difference in entropy of an individual strategy compared to the entropy of the whole population (Tuyls et al., 2003).

The dynamical model of Equation 10 assumes that all actions are updated simultaneously, as is the case for Cross learning. Q-learning, however, only updates the Q-value of the selected action, causing discrepancies between the predicted dynamics and the actual learning behaviour of the algorithm. The variation *frequency-adjusted Q-learning* (FAQ) (Kaisers & Tuyls, 2010) mimics simultaneous action updates by modulating the update rule (Equation 3) inversely proportional to x_i, thereby following the dynamical model of Equation 10 precisely. Dropping the state dependency, this yields

$$Q(i) \leftarrow Q(i) + \frac{1}{x_i}\alpha \left[r + \max_j Q(j) - Q(i)\right].$$

Using the replicator dynamics model of Equation 10, two independent proofs of convergence for FAQ have been derived for two-player two-action normal-form games, showing convergence near Nash equilibria given a decreasing exploration temperature τ (Kaisers & Tuyls, 2011; Kianercy & Galstyan, 2012).

Lenient FAQ (LFAQ) (Bloembergen et al., 2011) is a variation aimed at overcoming convergence to suboptimal equilibria by mis-coordination in the early phase of cooperative learning processes, when mistakes by one agent may lead to penalties for others, irrespective of the quality of their actions. Leniency towards such mistakes can be achieved by collecting κ rewards for each action, and updating the Q-value based on the highest of those rewards. This causes an (optimistic) change in the expected reward for the actions of the learning agent, incorporating the probability of a potential reward for that action being the highest of κ consecutive tries (Panait et al., 2008). The expected reward for each action $\mathbf{Ay}$ in

5. For a model of Boltzmann Q-learning dynamics with varying temperature, see the work of Kaisers, Tuyls, Parsons, and Thuijsman (2009) and Kaisers (2012).
6. From here on, we will derive the dynamics of one agent only. The dynamics of other agents follow straightforwardly, similar to Equation 9.

Equation 10 is replaced by the utility vector $\mathbf{u}$, with

$$u_i = \sum_j \frac{a_{ij} y_j \left[\left(\sum_{k:a_{ik} \leq a_{ij}} y_k \right)^\kappa - \left(\sum_{k:a_{ik} < a_{ij}} y_k \right)^\kappa \right]}{\sum_{k:a_{ik} = a_{ij}} y_k} \tag{11}$$

where $\sum_{k:\text{statement}}$ implies summing over all indices k for which the statement holds.

Recently, the evolutionary framework has also been extended to the polynomial weights algorithm, which implements *regret minimisation* (RM) (Blum & Mansour, 2007; Klos et al., 2010). The learner calculates the loss (or regret) l_i of taking action i rather than the best action in hindsight as $l_i = r^* - r$ where r is the actual reward received, and r^* is the optimal reward. The learner maintains a set of weights $\mathbf{w}$ for all actions, updates these weights according to the perceived loss, and derives a new policy by normalisation:

$$\begin{aligned} w_i &\leftarrow w_i \left[1 - \alpha l_i \right] \\ x_i &= \frac{w_i}{\sum_j w_j}. \end{aligned} \tag{12}$$

This form of regret, computed at each time step, is known as external regret, whereas internal regret is computed as the loss with respect to a policy that replaces a given action i with some other action j at each occurrence (Blum & Monsour, 2007). Despite the great difference in update rule and policy generation compared to Cross learning or Q-learning, Klos et al. (2010) show that the infinitesimal time limit of regret minimisation can similarly be linked to a dynamical system with replicator dynamics in the numerator:

$$\dot{x}_i = \frac{\alpha x_i \left[(\mathbf{A}\mathbf{y})_i - \mathbf{x}^\top \mathbf{A}\mathbf{y} \right]}{1 - \alpha \left[\max_k (\mathbf{A}\mathbf{y})_k - \mathbf{x}^\top \mathbf{A}\mathbf{y} \right]}. \tag{13}$$

The denominator can be interpreted as a learning rate modulation dependent on the best action's update, corresponding to weighting the action probability update by the expected loss.

4.1.2 Gradient Ascent Algorithms

Gradient ascent (or descent) is a well known optimisation technique in the field of Machine Learning. Given a well-defined differentiable objective function, the learning process can follow the direction of its gradient in order to find a local optimum. This concept can be adapted for multi-agent learning by having the learning agents' policies follow the gradient of their individual expected payoff. This approach assumes that the expected payoff function is known to (or can be accurately learned by) the agents, which may not generally be feasible in practice, since in multi-agent settings the payoff function usually depends on (possibly frequently changing) unobservable internal states of other agents.

One algorithm implementing gradient ascent for multi-agent learning is *infinitesimal gradient ascent* (IGA) (Singh et al., 2000), in which each learner updates its policy by taking infinitesimal steps in the direction of the gradient of its expected payoff. It has been proven that, in two-player two-action games, IGA either converges to a Nash equilibrium, or the asymptotic expected payoff of the two players converges to the expected payoff of a Nash equilibrium (Singh et al., 2000). A discrete time algorithm using a finite decreasing step size shares these properties. Take $V(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ to be the value function that maps a policy to its expected payoff. The policy update rule for IGA can now be defined as

$$\begin{aligned} \Delta x_i &\leftarrow \alpha \frac{\partial V(\mathbf{x})}{\partial x_i} \\ \mathbf{x} &\leftarrow \mathit{projection}(\mathbf{x} + \Delta \mathbf{x}) \end{aligned} \tag{14}$$

where α denotes the learning step size. The intended change $\Delta\mathbf{x}$ may take $\mathbf{x}$ outside of the valid policy space, in which case it is projected back to the nearest valid policy by the *projection* function.

Win or learn fast (IGA-WoLF) (Bowling & Veloso, 2002) is a variation on IGA which uses a variable learning rate. The intuition behind this scheme is that an agent should adapt quickly if it is performing worse than expected, whereas it should be more cautious when it is winning. The modified learning rule of IGA-WoLF is

$$\begin{aligned} \Delta x_i &\leftarrow \frac{\partial V(\mathbf{x})}{\partial x_i} \begin{cases} \alpha_{min} & \text{if } V(\mathbf{x}) > V(\mathbf{x}^*) \\ \alpha_{max} & \text{otherwise} \end{cases} \\ \mathbf{x} &\leftarrow \mathit{projection}(\mathbf{x} + \Delta \mathbf{x}) \end{aligned} \tag{15}$$

where $\mathbf{x}^*$ is a reference policy, e.g., a policy belonging to an arbitrary Nash equilibrium.

The *weighted policy learner* (WPL) (Abdallah & Lesser, 2008) is a second variation of IGA that also modulates the learning rate, but in contrast to IGA-WoLF it does not require a reference policy. The update rule of WPL is defined as

$$\begin{aligned} \Delta x_i &\leftarrow \alpha \frac{\partial V(\mathbf{x})}{\partial x_i} \begin{cases} x_i & \text{if } \frac{\partial V(\mathbf{x})}{\partial x_i} < 0 \\ 1 - x_i & \text{otherwise} \end{cases} \\ \mathbf{x} &\leftarrow \mathit{projection}(\mathbf{x} + \Delta \mathbf{x}) \end{aligned} \tag{16}$$

where the update is weighted either by x_i or by $1-x_i$ depending on the sign of the gradient. This means that $\mathbf{x}$ is driven away from the boundaries of the policy space.

In the next section, we will derive the dynamical model of these gradient ascent algorithms in two-player two-action games, and show their remarkable similarities to the dynamics of the reinforcement learners discussed in Section 4.1.1.

4.1.3 Comparative Overview of Learning Dynamics in 2x2 Games

For two-agent two-action games, the dynamical models presented in the previous sections can be simplified (Kaisers et al., 2012). Let $\mathbf{h} = (1, -1)$, $\mathbf{x} = (x, 1-x)$ and $\mathbf{y} = (y, 1-y)$. The learning dynamics are completely described by the pair $(\dot{x}, \dot{y})$, which denote the probability change of the first action for both learners. For Cross learning (CL) this leads to the simplified form

$$\begin{aligned} \dot{x} &= x\big[(\mathbf{A}\mathbf{y})_1 - \mathbf{x}^\top \mathbf{A}\mathbf{y}\big] \\ &= x(1-x)\big[y\,(a_{11} - a_{12} - a_{21} + a_{22}) + a_{12} - a_{22}\big] \\ &= x(1-x)\big[y\mathbf{h}\mathbf{A}\mathbf{h}^\top + a_{12} - a_{22}\big] \end{aligned}$$

where a_{12} and a_{22} are elements of the payoff matrix $\mathbf{A}$. To shorten the notation for two-action games, let

$$\eth = (\mathbf{A}\mathbf{y}^\top)_1 - (\mathbf{A}\mathbf{y}^\top)_2 = y\mathbf{h}\mathbf{A}\mathbf{h}^\top + a_{12} - a_{22}$$

denote the gradient, such that the CL dynamics are written as $\dot{x} = x(1-x)\eth$. Then, similarly, the simplified FAQ dynamics read

$$\dot{x} = \alpha x(1-x)\left[\frac{\eth}{\tau} - \log\frac{x}{1-x}\right].$$

The dynamics of RM are slightly more complex, as the denominator depends on which action gives the highest reward. This can be derived from the gradient: the first action will be maximal iff $\eth > 0$. Using this insight, the dynamics of RM in two action games can be written as follows:

$$\dot{x} = \alpha x(1-x)\eth \cdot \begin{cases} (1+\alpha x\eth)^{-1} & \text{if } \eth < 0 \\ (1-\alpha(1-x)\eth)^{-1} & \text{otherwise.} \end{cases}$$

For IGA, the update rule can be worked out in a similar fashion. The main term in this update rule is the gradient of the expected reward, which in two player two-action games can be written as

$$\begin{aligned} \frac{\partial V(x)}{\partial x} &= \frac{\partial}{\partial x}(x, 1-x)\mathbf{A}\begin{pmatrix} y \\ 1-y \end{pmatrix} \\ &= y(a_{11} - a_{12} - a_{21} + a_{22}) + a_{12} - a_{22} \\ &= y\mathbf{hAh}^\top + a_{12} - a_{22} \\ &= \eth. \end{aligned}$$

This reduces the dynamics of the update rule for IGA in two-player two-action games to $\dot{x} = \alpha\eth$. The extension of the dynamics of IGA to IGA-WoLF and WPL are straightforward.

Table 3 lists the dynamics of the six discussed algorithms: IGA, IGA-WoLF, WPL, CL, FAQ and RM. It is immediately clear from this table that all algorithms share the same basic term in their dynamics: the gradient $\eth$. Depending on the algorithm, the gradient is scaled with a learning speed modulation. Interestingly, the dynamics of IGA are completely independent of the learner's own current policy x. In other words, IGA is an

Table 3: This table shows an overview of the learning dynamics, rewritten for the specific case of two-agent two-action games (Kaisers et al., 2012).

Algorithm	$\dot{x}$
IGA	$\alpha\eth$
IGA-WoLF	$\eth \cdot \begin{cases} \alpha_{min} & \text{if } V(\mathbf{x}) > V(\mathbf{x}^*) \\ \alpha_{max} & \text{otherwise} \end{cases}$
WPL	$\alpha\eth \cdot \begin{cases} x & \text{if } \eth < 0 \\ (1-x) & \text{otherwise} \end{cases}$
CL	$x(1-x)\ \eth$
FAQ	$\alpha x(1-x)\left[\eth \cdot \tau^{-1} - \log\frac{x}{1-x}\right]$
RM	$\alpha x(1-x)\ \eth \cdot \begin{cases} (1+\alpha x\eth)^{-1} & \text{if } \eth < 0 \\ (1-\alpha(1-x)\eth)^{-1} & \text{otherwise} \end{cases}$

off-policy algorithm that assumes that all actions are sampled equally often. In contrast, in any multi-agent learning setting in which the (gradient of the) value function is not known, the learners necessarily need to be on-policy, as otherwise they cannot learn the true value function. In this light, it can be argued that Cross learning implements stochastic on-policy gradient ascent (Kaisers et al., 2012). Finally, FAQ yields the only dynamics that additionally add exploration terms to the process.

This analysis shows the merits of the evolutionary game theoretic approach to the study of multi-agent learning. By deriving mathematical models of the infinitesimal time limit of various learning algorithms, we can formally establish their underlying differences and commonalities.

4.2 Replicator Dynamics in Continuous Action Spaces

The dynamics and algorithms discussed previously assume a discrete, finite action space. In contrast, many real-world settings feature actions that are rather of a continuous nature. In such settings the tabular notation for Q-functions and policies is no longer feasible and *function approximators* need to be used. One approach is to discretise such continuous parameters in ranges, and treat each range as one instance. However, it is not straightforward in general to design a good discretisation, and details might be lost. Another approach is to model those continuous actions directly, moving from discrete probability vectors over actions to probability density functions. Examples of Q-learning algorithms for continuous action spaces are *fitted Q-iteration* (Ernst, Geurts, & Wehenkel, 2005) and *NEAT+Q-learning* (Whiteson & Stone, 2006). Learning automata can similarly be extended to continuous action spaces. For instance, Santharam, Sastry, and Thathachar (1994) propose *continuous action-set learning automata* (CALA) which use a Gaussian distribution to model the policy. It is also possible to use a nonparametric distribution, as is employed in *continuous action reinforcement learning automata* (CARLA), allowing a more diverse exploration strategy and convergence to a potentially multi-modal distribution (Howell, Frost, Gordon, & Wu, 1997; Rodríguez, Vrancx, Grau, & Nowé, 2012). An in-depth discussion of these learning methods falls outside the scope of this paper. Instead, we focus here on modelling such algorithms by extending the replicator dynamics to continuous action spaces.[7]

Suppose each agent's action space can be described by D continuous parameters $\mathbf{x} = (x_1, x_2, \dots, x_D)$, with $\mathbf{x} \in \Theta \subset \mathbb{R}^D$, where Θ is the allowed action space. In a two-player setting, the reward of playing action $\mathbf{x}$ against an opponent who plays $\mathbf{y}$ is given by $f(\mathbf{x}, \mathbf{y})$. An agent's policy at time t is now given by a probability density function over the action space, $\phi(\mathbf{x}, t)$, where

$$\int_\Theta \phi(\mathbf{x}, t) d\mathbf{x} = 1.$$

We can now write the continuum limit of the standard replicator dynamics (Equation 5) as a partial differential equation (Oechssler & Riedel, 2001; Cressman, 2005; Tuyls & Westra, 2009):

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \phi(\mathbf{x}, t)\Big[V(\mathbf{x}, t) - E(t)\Big] \tag{17}$$

7. A recent extensive overview of function approximation methods for reinforcement learning is provided by Busoniu et al. (2010).

where

$$V(\mathbf{x},t) = \int_{\Theta} f(\mathbf{x},\mathbf{y})\phi(\mathbf{y},t)d\mathbf{y}$$
$$E(t) = \int_{\Theta} V(\mathbf{x},t)\phi(\mathbf{x},t)d\mathbf{x}.$$

Here, $V(\mathbf{x})$ depicts the expected reward of action $\mathbf{x}$, and E the overall average reward. In the context of statistical mechanics these terms can be thought of as local potential and total energy, respectively.

Similar to the discrete action dynamics discussed before, we can add mutation terms to Equation 17 to model exploration by the learning agents. The most straightforward approach to do this is by including a diffusion term to the replicator dynamics, which allows the mutation of strategies to slightly different ones 'nearby'. This procedure was originally proposed by Hofbauer and Sigmund (1998), for which Ruijgrok and Ruijgrok (2005) have provided a rigorous mathematical foundation and subsequent analysis. Ruijgrok and Ruijgrok add such a diffusion term to the continuous action replicator dynamics and find that even a small mutation rate may greatly alter the outcome of the learning process, leading to more favourable results in, e.g., the ultimatum game. Specifically, they add a transition probability per unit time q_{ij} for the spontaneous transfer to strategy i from strategy j. In this way the discrete replicator equations with mutation can be written as follows:

$$\dot{x_i} = x_i(V_i - E) + \sum_j (q_{ij}x_j - q_{ji}x_i).$$

Ruijgrok and Ruijgrok follow the method of Van Kampen (1992) to derive the continuum limit of the mutation term. Using this approach, they arrive at the same continuous replicator equation as Hofbauer and Sigmund:

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = \phi(\mathbf{x},t)\Big[V(\mathbf{x},t) - E(t)\Big] + \mu(\mathbf{x},t)\nabla^2\phi(\mathbf{x},t). \tag{18}$$

Ruijgrok and Ruijgrok study this equation for a number of games that have been transformed to a continuous strategy setting. There is, however, no reason to assume that mutations are only restricted to adjacent strategies. Therefore, building on this work, Tuyls and Westra (2009) compare three different diffusion-based mutation terms, and find that the type of mutation can also significantly influence the resulting dynamics. Specifically, they replace the simple isotropous diffusion term containing the mutation rate of Ruijgrok and Ruijgrok by mimicking the effect of anisotropous deterministic mutations. They start from the continuous time discrete action replicator equations with mutation, and from there derive a different formulation for the continuous action space:

$$\frac{\partial \phi(\mathbf{x},t)}{\partial t} = \phi(\mathbf{x},t)\Big[V(\mathbf{x},t) - E(t)\Big] + \nabla^T \cdot \mu(\mathbf{x})\nabla\Big[V(\mathbf{x})\phi(\mathbf{x},t)\Big].$$

This new formulation is able to capture certain aspects of an evolution driven by game theoretic interactions that are absent in the original formulation of Hofbauer and Sigmund, and Ruijgrok and Ruijgrok in Equation 18. The dynamics of this approach are even more complex as different mutation rates provide entirely different stationary solutions. Full details can be found in the work of Tuyls and Westra.

Although the dynamical models of Equations 17 and 18 have not yet been directly linked to a specific learning algorithm, Galstyan (2013) recently proposed a dynamical model for a continuous action version of Boltzmann Q-learning. Galstyan's model has a similar selection term based on local potential $V(\mathbf{x})$ and total energy E, but a mutation term that is again an entropy term derived from the Boltzmann distribution, as in Equation 10. Galstyan finds that mutation drives the learning process away from pure Nash equilibria but helps convergence to uniformly mixed equilibria, similar to discrete action Q-learning (see Kaisers & Tuyls, 2011; Kianercy & Galstyan, 2012).

4.3 State-Coupled Replicator Dynamics

So far, our discussion of learning dynamics has been limited to stateless games. Although many real-world interactions can indeed be cast in the form of repeated normal-form games, this is not always the case. Therefore, there is a need to understand learning dynamics in statefull environments as well. Vrancx et al. (2008a) propose a combination of replicator dynamics and switching dynamics to model learning automata in stochastic (Markov) games. As learning automata are by definition stateless (see Section 3.1), an extension is needed for multi-state games. One option is for an agent to maintain a *network of learning automata* (Wheeler Jr & Narendra, 1986; Vrancx et al., 2008b), one for each state. As the game progresses, control is passed from one automaton to the other depending on the current state of the game. Instead of performing policy updates for the active automaton based on the immediate reward r_t, the update is delayed until the automaton becomes active again, at which time it is updated based on the average reward received during that period. Based on this mechanism, Vrancx et al. (2008a) partition the state-space in cells, corresponding to different attractors in these average reward games. Each cell has a fixed replicator dynamic, based on which the agents' policies are updated. When this update makes the system leave the current cell, a new replicator dynamic takes over. Hennes, Tuyls, and Rauterberg (2008) further formalise these *piece-wise replicator dynamics*.

The piece-wise model suffers from several shortcomings, however, as demonstrated by Hennes et al. (2009). Firstly, the model only approximates the learning behaviour by assuming fixed dynamics in each cell, and secondly, the discrete switching between cells causes discontinuities that are not present in real traces of the learning process. In order to alleviate these shortcomings, Hennes et al. propose the *state-coupled replicator dynamics* (SC-RD) model which uses direct state coupling rather than piece-wise switching dynamics. The direct state coupling eliminates anomalies and discontinuities caused by linear approximations and discrete cell switching. The SC-RD are defined as follows. Let $\boldsymbol{\pi}$ be the set of policies of n agents, i.e., $\boldsymbol{\pi} = \left(\pi^1, \pi^2, \ldots, \pi^n\right)$. Moreover, let $\mathbf{a} = \left(a^1, a^2, \ldots, a^n\right)$ be their joint action. Assuming the game has no absorbing states (i.e., the set of states S is ergodic), there exists a stationary distribution $\chi^{\boldsymbol{\pi}}$ over all states S under $\boldsymbol{\pi}$, where $\chi^{\boldsymbol{\pi}}_s$ is the frequency of state s and $\sum_{s\in S} \chi^{\boldsymbol{\pi}}_s = 1$. We can then calculate the limiting average reward $\bar{r}$ of playing a specific joint-action $\mathbf{a}$ in state s, given fixed policies $\boldsymbol{\pi}$ in all other states s', as

$$\bar{r}^i(s, \mathbf{a}) = \chi^{\boldsymbol{\pi}}_s r^i(s, \mathbf{a}) + \sum_{s'\in S-\{s\}} \chi^{\boldsymbol{\pi}}_{s'} f^i\left(s'\right) \tag{19}$$

where $f^i(s')$ is the expected reward (fitness) of agent i in state s', calculated as

$$f^i(s) = \sum_{\mathbf{a} \in \prod_{j=1}^n A^j} \left(r^i(s, \mathbf{a}) \prod_{k=1}^n \pi^k \left(s, a^k\right) \right).$$

We can set up a system of differential equations for each agent i and action j, similar to Equation 9, where the payoff matrix $\mathbf{A}$ is substituted by the limiting average reward $\bar{r}$. Furthermore, instead of the single opponent policy σ we now have all other agents' policies $\boldsymbol{\pi}^{-i} = \left(\pi^1 \dots \pi^{i-1}, \pi^{i+1} \dots \pi^n\right)$. The expected payoff for player i playing pure action j in state s is given by

$$f_j^i(s) = \sum_{\mathbf{a}' \in \prod_{l \neq i} A^l} \left(\bar{r}^i(s, \mathbf{a}') \prod_{k \neq i} \pi^k \left(s, a'^k\right) \right)$$

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

$$f^i(s, \omega) = \sum_{a^i \in A^i} \left[\omega(s, a^i) \sum_{\mathbf{a}' \in \prod_{l \neq i} A^l} \left(\bar{r}^i(s, \mathbf{a}') \prod_{k \neq i} \pi^k \left(s, a'^k\right) \right) \right]$$

where $\bar{r}$ is the limiting average reward given in Equation 19. Writing $\mathbf{x}^i(s) \equiv \pi^i(s)$ to be the probability distribution over actions of agent j in state s, we can now define the multi-population state-coupled replicator dynamics as the following system of differential equations:

$$\dot{x}_j^i(s) = x_j^i(s) \chi_s^{\mathbf{x}} \left[f^i(s, e_j) - f^i\left(s, \mathbf{x}^i(s)\right) \right] \tag{20}$$

where e_j is the j^{th}-unit vector, corresponding to the policy that plays pure action j. In total this system has $N = \sum_{s \in S} \sum_{i=1}^n |A^i|$ replicator equations. Hennes et al. show that the SC-RD model describes the true learning dynamics of networks of learning automata far more precise than the piece-wise replicator dynamics.

Recently, the replicator dynamics have been extended to the sequence form representation of *extensive-form games* as well (Gatti, Panozzo, & Restelli, 2013; Lanctot, 2014). This allows us to model even more complex games, e.g. with sequential moves and imperfect information. Panozzo et al. (2014) have developed a new version of Q-learning that works on the sequence form, along with a dynamical model that matches the learning process based on the sequence-form replicator dynamics with a mutation term similar to Equation 10. They show that, although the selection mechanism of the sequence-form and normal-form replicator dynamics are realization equivalent (Gatti et al., 2013; Lanctot, 2014), the mutation term is not. An in-depth discussion of these findings falls outside the scope of this article, as it would require the addition of a much broader background on extensive-form games. However, these recent works show the promise of the evolutionary framework for multi-agent learning beyond normal-form or even stochastic games as well.

5. Experimental Overview

We established the link between multi-agent reinforcement and the replicator dynamics of evolutionary game theory in Section 3, and provided an overview of learning dynamics in normal-form games, continuous strategy spaces, and stochastic (Markov) games in Section 4. Here, we show a set of experiments that empirically validate these models in two-player two-action normal-form games. We restrict ourselves to these games as their simplicity allows easy visual analysis, while preserving the explanatory power of such dynamical models. At the end of this section we provide an overview of related empirical work in more complex interactions.

As described earlier, in two-player two-action games the policy space can be compactly represented by the unit simplex as it is completely defined by the probability with which both agents select their first action. This makes it easy to plot and visually inspect the learning dynamics in such interactions. In Section 3.1, Figure 5, an example of such analysis can be found for Cross learning, as compared to the standard replicator dynamics. Similar analysis can be performed for different learning algorithms, by plotting policy traces of their learning behaviour overlaid on the vector field of their corresponding dynamical model (see e.g. Tuyls et al., 2006; Klos et al., 2010; Kaisers & Tuyls, 2010, 2011; Bloembergen et al., 2011). Figure 6 shows this for standard frequency-adjusted Q-learning (FAQ), and its lenient counterpart (LFAQ), in the top two rows. Whereas the dynamics of these different algorithms are similar in their convergence behaviour when only one equilibrium is present, as is the case in the prisoner's dilemma and matching pennies, in the stag hunt differences can be observed. The notion of leniency, introduced to overcome convergence to suboptimal equilibria, works to drive the learning process towards the optimal outcome of the game (S, S – top right corner), as can be easily observed by investigating the vector field. Interestingly, while Cross learning and the standard replicator dynamics do not converge in the matching pennies game, (L)FAQ does sprial inwards towards the single Nash equilibrium at $(\frac{1}{2}, \frac{1}{2})$, which is not evolutionarily stable in the classical replicator dynamics model (Section 2.3). The additional exploration term makes a difference here.

So far, we have analysed the dynamics of two identical learners pitted against each other. However, the replicator dynamics model allows for heterogeneous systems as well, in which different agents follow different learning rules. In such cases, the policy change of each individual agent is modelled by a different variation of the replicator dynamics, corresponding to that agent's learning rule. The bottom row of Figure 6 shows this for the situation where FAQ and LFAQ are pitted against each other. In games where the self-play dynamics of both learners are similar, such as the prisoner's dilemma and matching pennies, the mixed dynamics do not change significantly. In other cases, such as the stag hunt, the learning process is clearly influenced as different dynamics mix. As LFAQ has a stronger tendency to play the optimal action S, FAQ is persuaded to do likewise. Such analysis makes it possible to compare the behaviour of learning algorithms in heterogeneous environments, or to compare different parameter settings of a single algorithm.

As a final example, we revisit the comparison of gradient ascent-based and reinforcement learning algorithms of Section 4.1.3. Figure 7 shows the learning dynamics as predicted by the models derived there and presented in Table 3, for the matching pennies game. Regret minimisation (RM) is omitted as its dynamics are visually indistinguishable from Cross

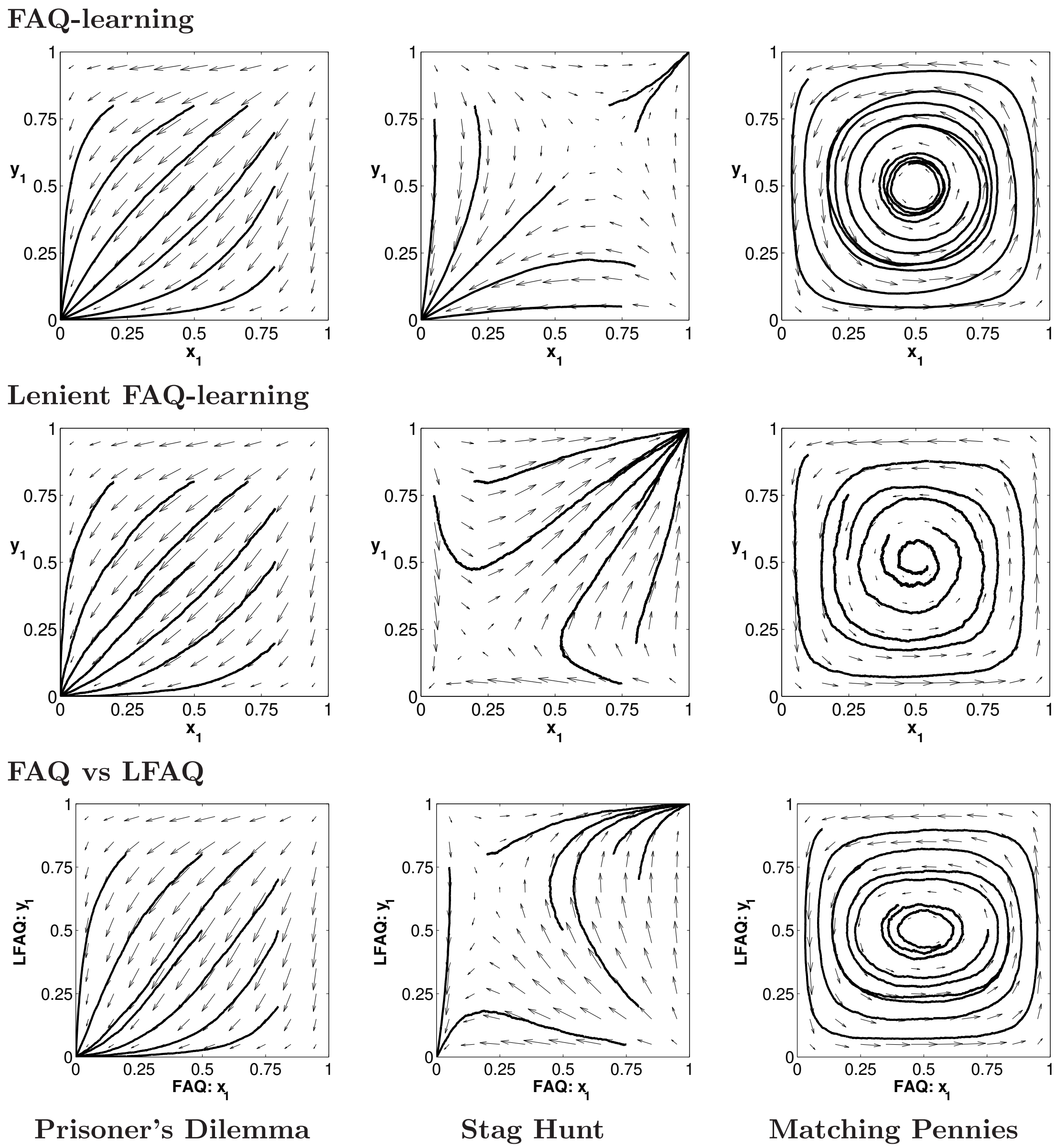


Figure 6: Policy traces of FAQ and LFAQ, plotted in the unit simplex and overlaid on their respective dynamical model, for the prisoner's dilemma (left), the stag hunt (center), and matching pennies (right) (Bloembergen et al., 2011).

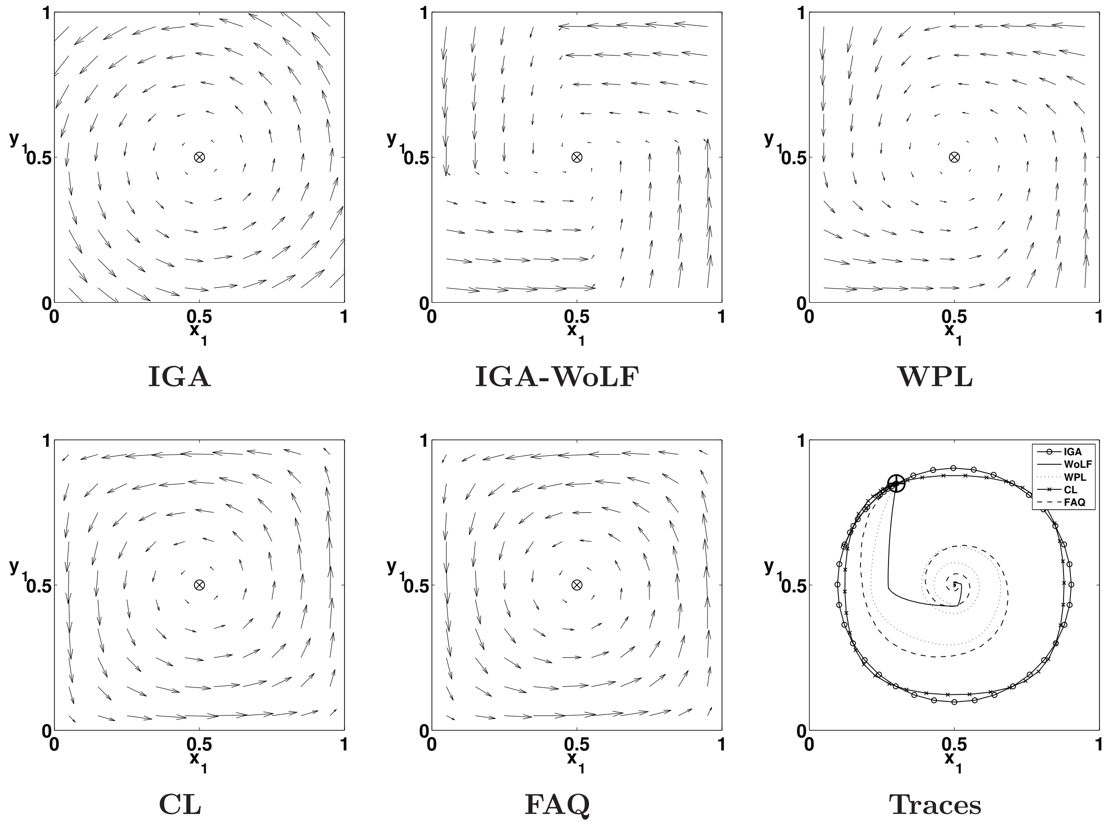


Figure 7: Overview of the dynamics of various gradient ascent and reinforcement learning algorithms in matching pennies. The bottom right panel shows a single trace of the dynamical models, using the same initial policy (indicated with $\oplus$) (Kaisers et al., 2012).

learning (CL). We can clearly observe the similarity between CL and infinitesimal gradient ascent (IGA), which both cycle around the central equilibrium point without converging. Win-or-learn-fast IGA (WoLF) and the weighted policy learner (WPL) both converge due to their learning rate modulator; in the case of WoLF, two learning rates are used (for winning and loosing), whereas WPL uses a continuum of learning rates, resulting in non-linear dynamics. This difference can be clearly observed by comparing the vector fields of their respective dynamical models. As noted before, FAQ spirals inwards towards the Nash equilibrium – although this is hard to observe from the vector field alone, this fact can be verified by following a trace of the dynamics, as shown in the bottom right panel of Figure 7 for the different algorithms. These traces highlight the (subtle) similarities and differences between the diverse algorithms.

Similar analyses have been performed to investigate learning dynamics in more complex (e.g., multi-state) games, or to highlight the influence of certain parameters on the learning process. For instance, Tuyls et al. (2003) were the first to derive the selection-mutation dynamics of Q-learning in discrete-action normal-form games, and to visually compare traces of the learning algorithm with the predicted dynamics of the model. Similar derivations and analyses have later been provided for other algorithms such as learning automata (Tuyls et al., 2006) and regret minimisation (Klos et al., 2010). Others have focused on learning dynamics in multi-state (stochastic) games, in particular deriving dynamics of networks of learning automata (Vrancx et al., 2008a; Hennes et al., 2009). Again others have derived new learning algorithms based on a desirable dynamical model, leading to FAQ-learning (Kaisers & Tuyls, 2010) and lenient FAQ (Bloembergen et al., 2011) for normal-form games, and RESQ-learning for stochastic games (Hennes et al., 2010). Finally, several authors have used insights stemming from the dynamical models to compare different algorithms, or to investigate their convergence. For example, Bowling and Veloso (2002) introduce the variable learning rate WoLF heuristic and prove that it can be used to make IGA convergent in normal-form games. Abdallah and Lesser (2008) introduce WPL and compare it to IGA and IGA-WoLF, and show that WPL converges in normal-form games as well, without requiring as much information as IGA-WoLF. Kaisers and Tuyls (2011) use the dynamical model of FAQ to demonstrate its near-NE convergence in normal-form games, and Galstyan (2013) similarly shows this for continuous-action Q-learning (see also Section 6.1.2).

Table 4 presents an overview of these related works, clustered by interaction type – i.e., normal form games with discrete or continuous actions spaces, and stochastic games – and lists the relevant learning algorithms that are investigated. We purposely list only those works that explicitly focus on the relation between multi-agent learning algorithms and the evolutionary dynamical model. Each of these works utilises and extends this connection in order to gain qualitative insights into the behaviour of such algorithms in a variety of settings.

Table 4: An overview of related empirical evaluations of learning dynamics. NFG: normal-form games; CNFG: continuous action normal-form games; SG: stochastic (Markov) games.

Type	**Algorithm**	**Reference**
NFG	Q-learning	Tuyls et al. (2003, 2006)
NFG	regret minimisation	Klos et al. (2010)
NFG	FAQ	Kaisers and Tuyls (2010, 2011)
NFG	lenient FAQ	Bloembergen et al. (2011) Kaisers (2012)
NFG	WoLF	Bowling and Veloso (2002)
NFG	IGA, IGA-WoLF, WPL	Abdallah and Lesser (2008)
CNFG	Q-learning	Galstyan (2013)
SG	networks of learning automata	Vrancx et al. (2008a) Hennes et al. (2009)
SG	RESQ-learning	Hennes et al. (2010)

6. Applications

In the previous sections we highlighted the descriptive power of the replicator dynamics model of multi-agent learning. Here, we focus on its prescriptive power as well. For example, we can use the dynamical models for easy parameter tuning of learning algorithms. Moreover, starting from desired dynamics, it is possible to reverse-engineer a learning algorithm that exhibits the preferred behaviour. Finally, the evolutionary models can be used to analyse complex strategic interactions, such as automated trading. Focusing on meta-strategies rather than primitives reduces the complexity of such interactions enough to study their dynamics analytically.

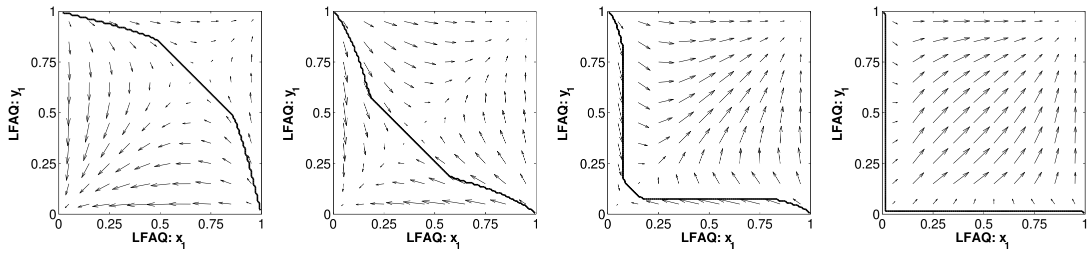
6.1 Parameter Tuning

Parameter tuning is traditionally a cumbersome task involving many simulation trials, often following some evolutionary optimisation approach. However, with a deterministic dynamical model the effect of various parameters on the learning process is readily observable. In the following, we provide examples for lenient learning, and for balancing exploration and exploitation.

6.1.1 Degree of Leniency

Lenient learning was introduced as a way to overcome the problem of suboptimal convergence in cooperative multi-agent settings, where initial mis-coordination leads to an undervaluation of the optimal action (Panait et al., 2008). This problem is also known as *relative overgeneralisation* (Wiegand, 2003) or *action shadowing* (Fulda & Ventura, 2007). By focusing the learning update on maximal rewards rather than average, the learner effectively ignores the low rewards that are due to suboptimal behaviour by others in the early phases of the learning process. This can be achieved by collecting κ rewards for each action before performing an update based on the highest of those rewards (Panait et al., 2008; Bloembergen et al., 2011). The details and dynamical model of lenient frequency-adjusted Q-learning (LFAQ) are given in Section 4.1.1, Equation 11.

The main parameter of LFAQ is the degree of leniency, κ. One of the main advantages of having a dynamical model is that it allows studying and tuning such a parameter without the need for extensive simulations. Instead, we can directly analyse the dynamical model. Figure 8 shows the dynamics of LFAQ, following Equation 11, in the stag hunt, for $\kappa \in \{1, 2, 5, 25\}$. The stag hunt has two pure Nash equilibria that are also evolutionarily stable strategies, where both agents either play H, at $(0, 0)$, or S, at $(1, 1)$ (see Section 2.3). The dilemma in this game is the choice between the safe action H, which always gives the same reward independent of the other agent's action, or S, which is optimal, but only if both players coordinate. This is precisely the problem that leniency aims to solve. Figure 8 shows that as the degree of leniency increases, so does the basin of attraction of the optimal outcome $(1, 1)$. Depending on the nature of the game and the opponent, a balance needs to be found between leniency on the one hand, and the risk of being exploited, or lagging behind a changing environment, on the other. The dynamical model greatly facilitates gaining quick insight into these effects.



$\kappa = 1$ $\kappa = 2$ $\kappa = 5$ $\kappa = 25$

Figure 8: The effect of the degree of leniency κ on convergence in the stag hunt game. The solid line indicates the boundary between the basins of attraction for the two equilibria; the global optimum is located at $(1, 1)$ (Bloembergen et al., 2011).

6.1.2 Tuning the Exploration Rate in FAQ-Learning

Balancing exploration and exploitation is of vital importance to any learning task, in particular in dynamic environments where multiple learning agents interact. In (FA)Q-learning with Boltzmann exploration (Equation 4), the temperature parameter τ controls the level of exploration – a high temperature promotes exploration, whereas a low temperature favours exploitation. The dynamical model of FAQ (Equation 10) allows to study the effect of the exploration rate on the behaviour and convergence of the learner in a multi-agent setting, as demonstrated by Kaisers and Tuyls (2011).

Figure 9 shows the effect of τ on the dynamics and convergence of FAQ in battle of the sexes. In this game, players need to coordinate on either one of their two actions, however both have a different preference over those joint outcomes:

$$\begin{array}{cc} & \begin{array}{cc} B & S \end{array} \\ \begin{array}{c} B \\ S \end{array} & \begin{pmatrix} 2,1 & 0,0 \\ 0,0 & 1,2 \end{pmatrix} \end{array}$$

The first three frames of the figure show the dynamics and computed fixed points for different temperatures (first frame $\tau = \infty$, second frame $\tau = 0.72877$, third frame $\tau = 0$). The fixed points move between these discrete values for τ as indicated by the paths shown in the fourth frame. The game has three Nash equilibria, at $(0, 0)$, $(1, 1)$, and $(\frac{2}{3}, \frac{1}{3})$. However, for high values of τ the replicator model yields only one attracting fixed point, that moves from $(\frac{1}{2}, \frac{1}{2})$ towards the mixed equilibrium $(\frac{2}{3}, \frac{1}{3})$. Kaisers and Tuyls show that this fixed point splits in a supercritical pitchfork bifurcation at the critical temperature $\tau_{crit} \approx 0.72877$ and at position $(x_1, y_1) \approx (0.5841, 0.4158)$. For low temperatures $\tau < \tau_{crit}$, the dynamics yield three fixed points that move closer to the corresponding equilibria as τ is decreased. The two fixed points moving toward the pure equilibria $(0, 0)$ and $(1, 1)$ are attracting, and the third one moving toward $(\frac{2}{3}, \frac{1}{3})$ is repelling.

A similar analysis for Boltzmann Q-learning has been performed by Kianercy and Galstyan (2012). They also observe that the fixed points of the dynamical model move towards Nash equilibria as $\tau \to 0$ and relate the temperature of the Boltzmann mechanism to the thermodynamical concept of free energy (Galstyan, 2013). Gomes and Kowalczyk (2009) propose a dynamical model of ϵ-greedy Q-learning, and show that this model accurately

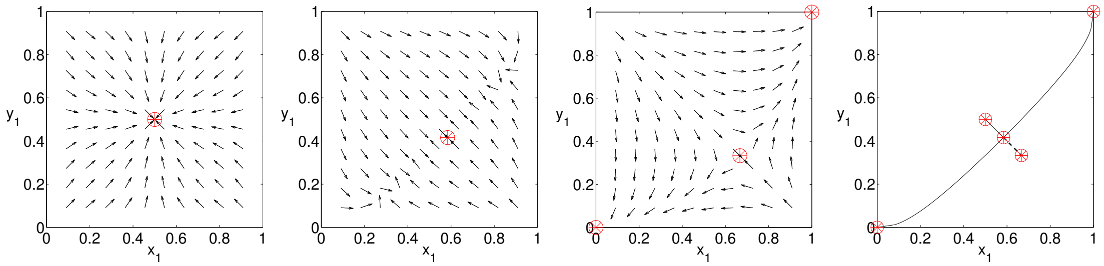


Figure 9: Replicator dynamics (arrows) and fixed points ($\otimes$) for $\tau = \{\infty, 0.72877, 0\}$ (first three frames). Right-most frame shows trajectories of fixed points as temperature is decreased (Kaisers & Tuyls, 2011).

predicts the empirical findings. Similarly, Wunder, Littman, and Babes (2010) present a detailed study of ϵ-greedy Q-learning in various classes of normal-form games and provide proofs of (non-) convergence for each class, varying from rapid convergence to stable oscillations. Each of these works shows the great applicability and benefit of the replicator dynamics model of multi-agent learning, when investigating the effect of various parameters on the learning process.

6.2 Design of New Learning Algorithms

So far, we have taken a *forward* approach: starting from a learning algorithm we derived a dynamical model that accurately predicts the behaviour of that algorithm in the limit. We can, however, also take an *inverse* approach by starting from a set of desired dynamics and reverse-engineering a learning algorithm that exhibits those dynamical properties (Hennes et al., 2010; Tuyls, Heytens, Nowé, & Manderick, 2003).

As an example, consider again the state-coupled replicator dynamics (SC-RD) introduced in Section 4.3. These dynamics describe the behaviour of networks of learning automata, which are essentially exploiting, and exploration is solely induced by the stochastic action-selection process. However, results from the domain of stateless games suggest that exploration aids convergence to mixed equilibria, where purely exploitative learners enter cycles (see Section 5, in particular Figure 7). Hennes et al. (2010) extend the SC-RD model (Equation 20) with the exploration term of FAQ-learning (Equation 10), which leads to

$$\dot{x}^i_j(s) = x^i_j(s)\chi^{\mathbf{x}}_s\Bigg[\Big[f^i\left(s, e_j\right) - f^i\left(s, \mathbf{x}^i(s)\right)\Big] \\ - \tau\Bigg(\log x^i_j + \sum_k x^i_k \log x^i_k\Bigg)\Bigg].$$

These dynamics can be translated to a learning algorithm by adding a similar exploration term to the policy update of a network of learning automata. The reward remains equal to the average accumulated reward since the last visit to that particular automata, while the policy update after taking action j is now computed as

$$\pi(i) \leftarrow \pi(i) + \alpha \begin{cases} \bar{r} - \pi(i)\bar{r} - \tau \left(\log \pi(i) + \sum_k \pi(k) \log \pi(k)\right) & \text{if } i = j \\ -\pi(i)\bar{r} - \tau \left(\log \pi(i) + \sum_k \pi(k) \log \pi(k)\right) & \text{otherwise.} \end{cases}$$

Hennes et al. show that RESQ-learning is able to converge in a two-state version of matching pennies, where the standard SC-RD cycle around the equilibrium.

Another example is given in the work of Tuyls et al. (2003), where the standard replicator dynamics are extended to ensure stable convergence to Nash equilibria in all classes of two-agent two-action normal-form games. Based on these extended replicator dynamics Tuyls et al. derive an extended Cross learning algorithm that adheres to the preferred dynamics.

6.3 Evolutionary Analysis of Complex Strategic Interactions

In addition to relatively simple, stylised games, we also can analyse much more complex systems. This is accomplished by taking a high-level view and focusing on meta-strategies, rather than atomic actions, in such scenarios. For example, this allows us to study the evolutionary dynamics of various trading strategies in stock markets (Kaisers et al., 2009; Hennes, Bloembergen, Kaisers, Tuyls, & Parsons, 2012; Bloembergen, Hennes, McBurney, & Tuyls, 2015). Similarly, it is possible to compare auction mechanisms (Phelps, Parsons, & McBurney, 2005), strategies in the game of poker (Ponsen, Tuyls, Kaisers, & Ramon, 2009), or even collision avoidance methods in multi-robot systems (Hennes, Claes, & Tuyls, 2013). Moreover, the link between the replicator dynamics and reinforcement learning allows us to predict what will happen when agents *learn* to optimise their strategy in such scenarios.

In the following, we present two examples of such analyses. Firstly, we detail *heuristic payoff tables* as a method to estimate the payoff of high-level meta-strategies empirically. Next, we present how to use this analysis to evaluate trading strategies in stock markets (Section 6.3.2), and collision avoidance strategies in multi-robot systems (Section 6.3.3).

6.3.1 Heuristic Payoff Tables

In order to analyse the evolutionary strength of high-level meta-strategies, we need to estimate the expected payoff of such strategies relative to each other. In evolutionary game theoretic terms, this is the relative fitness of the various strategies, dependent on the current frequencies of those strategies in the population. The evolutionary model assumes an infinite population. We cannot compute the payoff for such a population directly, but we can approximate it from evaluations of a finite population. All possible distributions over k strategies can be enumerated for a finite population of n individuals. Let N be a matrix, where each row N_i contains one discrete distribution. The matrix will yield $\binom{n+k-1}{n}$ rows. Each distribution over strategies can be simulated (using an appropriate model of the environment), returning a vector of expected relative rewards $u(N_i)$. Let U be a matrix which captures the rewards corresponding to the rows in N, i.e., $U_i = u(N_i)$. A heuristic payoff table $H = (N, U)$ is proposed by Walsh, Das, Tesauro, and Kephart (2002) to capture the payoff information for all possible discrete distributions in a finite population. An example of such a heuristic payoff table is given in Table 5. In this example, we have $k = 3$ different meta-strategies, distributed over a population of $n = 6$ individuals. Each row in N specifies exactly how many individuals use each of the three strategy types, and

Table 5: Example of a heuristic payoff table for $k = 3$ strategies and a finite population of $n = 6$ individuals.

N_{i1}	N_{i2}	N_{i3}	U_{i1}	U_{i2}	U_{i3}
6	0	0	0.5	0	0
5	1	0	0.4	0.7	0
	...			...	
3	1	2	0.3	0.5	0.8
	...			...	
0	0	6	0	0	0.9

each row in U specifies their estimated payoff. If a discrete distribution N_i features zero individuals of type j, their payoff naturally cannot be measured, and we set $U_{ij} = 0$.

In order to approximate the payoff for an arbitrary mix of strategies in an infinite population distributed over the species according to $\mathbf{x}$, n individuals are drawn randomly from the infinite distribution. The probability for selecting a specific row N_i can be computed from $\mathbf{x}$ and N_i as

$$P(N_i|\mathbf{x}) = \binom{n}{N_{i1}, N_{i2}, \ldots, N_{ik}} \prod_{j=1}^{k} x_j^{N_{ij}}.$$

The expected payoff of strategy i, $f_i(\mathbf{x})$, is then computed as the weighted combination of the payoffs given in all rows:

$$f_i(\mathbf{x}) = \frac{\sum_j P(N_j|\mathbf{x}) U_{ji}}{1 - (1 - x_i)^k}.$$

This expected payoff function can be used in Equation 5 to compute the evolutionary population change according to the replicator dynamics.

6.3.2 The Value of Information in Markets

As an example, consider a market in which differently informed traders bid for a certain asset (Bloembergen et al., 2015). Depending on their information level, traders have a certain amount of foresight regarding the future value of the stock. More information gives a better approximation of the real current value of the asset; however, information comes at a price. We can use the replicator dynamics model to analyse the effect of various pricing schemes for information, e.g. no cost, fixed cost for any amount of information, or a cost function that is linear in the amount of information. Traders can also choose not to acquire additional information but instead rely solely on the current market price. By running market simulations with different distributions of information levels among the traders, we can compute a heuristic payoff table as described above, and use this table as basis for the replicator model.

Figure 10 shows the resulting dynamics for three types of traders (uninformed – ZI; averagely informed – I3; insiders – I9), and the three different cost functions described above. In the absence of costs, the best strategy is to obtain as much information as

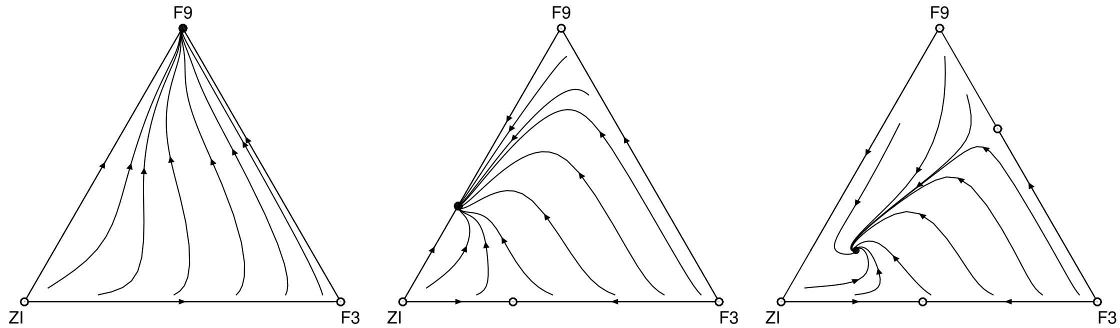


Figure 10: Vector field showing the evolutionary dynamics of a market with three information levels and different cost functions for information: no cost (left), fixed cost (center), and linear cost (right) (Bloembergen et al., 2015).

possible, leading to a domination of I9 traders over the entire interior of the simplex. Adding fixed costs gives a small boost to the I0 traders (who do not have to pay), allowing them to survive in equilibrium alongside the insiders. The linear cost function gives rise to an internal equilibrium where all types can coexist. Such analysis can be a valuable tool to gain insight into the dynamics of stock markets, and helps to predict the effect of external influences (such as costs, e.g. in the form of taxing schemes) on the market as a whole.

6.3.3 Collision Avoidance in Multi-robot Systems

Autonomous collision avoidance is a complex task in the field of robotics, especially in the presence of dynamic obstacles. The task increases in complexity when the dynamic obstacles are mobile robots that also take actions to avoid collisions. However, assuming mutual avoidance (reciprocity) may potentially improve avoidance behaviour since each robot only needs to take half of the responsibility of avoiding pairwise collisions. In order to test this hypothesis we can employ the same meta-strategy approach to evaluate the evolutionary strength of different collision avoidance strategies (Hennes et al., 2013).

One approach to collision avoidance in continuous spaces is the velocity obstacle (VO) paradigm, first introduced by Fiorini and Shiller (1998) for local collision avoidance and navigation in dynamic environments with multiple moving objects. The VO strategy can be modified to include reciprocity, yielding the reciprocal velocity obstacle (RVO); a hybrid between these two is given by the HRVO. Figure 11 (left) shows the evolutionary dynamics resulting from a heuristic payoff table over these three strategies. All pure population states are asymptotically stable fixed points under the replicator dynamics. Although the basin of attraction for RVO is considerably smaller, we do not see any clearly dominant strategy in this setting. For pairwise comparison between two strategies, along the faces of the simplex, no strategy is inferior; all three strategies are evolutionarily stable.

The VO takes all obstacles into account, independent of their distance. This may greatly reduce the mobility of robots in a highly cluttered environment. In order to overcome this problem, the VO can be truncated to ignore obstacles that are farther away. Truncation yields significantly different and more complex dynamics, as shown in Figure 11 (right). In a pairwise comparison (faces of the simplex), RVO is dominated by VO as well as by HRVO.

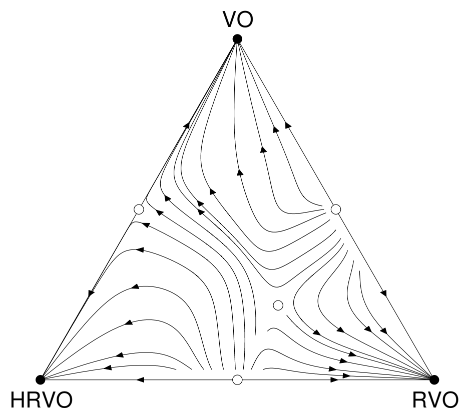


Figure 11: Evolutionary dynamics of three strategies for multi-robot collision avoidance, without truncation (left) and with truncation (right) (Hennes et al., 2013).

However, the reciprocal velocity obstacle is more robust in the presence of all three strategies (interior of the simplex), which can be explained by it being the more "aggressive", or the least restricting, strategy.

7. Conclusions

In this article we have surveyed recent advances in the study of the evolutionary dynamics of multi-agent learning. In particular, we presented the formal relation between reinforcement learning and the replicator dynamics of evolutionary game theory. By modifying the standard replicator dynamics, the behaviour of various state-of-the-art reinforcement learning algorithms in a multi-agent setting can be modelled accurately. So far, this link has been established in stateless environments (e.g. normal form games), both with discrete and continuous action spaces, and multi-state environments (e.g. stochastic games) with a discrete action space. Therefore, an important avenue for future work is the extension of the theory to stochastic games with continuous action spaces.

The analytical study of multi-agent learning dynamics offers several important advantages. In particular, it sheds light into the black box of reinforcement learning, by making it possible to analyse the learning dynamics of multi-agent systems in detail, and to compare the behaviour of different algorithms in a principled manner. This in turn facilitates important tasks such as parameter tuning. Furthermore, studying the dynamics of different learning algorithms helps in selecting a specific learner for a given problem. Moreover, it is possible to derive new learning algorithms by first designing preferred dynamics. Finally, the theory can be applied to complex strategic interactions in real-world settings by analysing meta-strategies, as demonstrated for automated trading and multi-robot collision avoidance.

Acknowledgments

We are grateful to the editor and anonymous reviewers of JAIR for their valuable feedback and helpful suggestions.

References

Abdallah, S., & Kaisers, M. (2013). Addressing the policy-bias of Q-learning by repeating updates. In *Proc. of the 2013 int. conf. on Autonomous Agents and Multi-Agent Systems (AAMAS 2013)*, pp. 1045–1052. International Foundation for AAMAS.

Abdallah, S., & Lesser, V. (2008). A multiagent reinforcement learning algorithm with non-linear dynamics. *Journal of Artificial Intelligence Research*, *33*(1), 521–549.

Agogino, A. K., & Tumer, K. (2012). A multiagent approach to managing air traffic flow. *Autonomous Agents and Multi-Agent Systems*, *24*(1), 1–25.

Ahmadi, M., & Stone, P. (2006). A multi-robot system for continuous area sweeping tasks. In *ICRA*, pp. 1724–1729.

Axelrod, R., & Hamilton, W. D. (1981). The evolution of cooperation. *Science*, *211*(4489), 1390–1396.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

Bloembergen, D., Hennes, D., McBurney, P., & Tuyls, K. (2015). Trading in markets with noisy information: An evolutionary analysis. *Connection Science*, *to appear*.

Bloembergen, D., Kaisers, M., & Tuyls, K. (2011). Empirical and theoretical support for lenient learning. In Tumer, Yolum, Sonenberg, & Stone (Eds.), *Proc. of 10th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pp. 1105–1106. International Foundation for AAMAS.

Blum, A., & Mansour, Y. (2007). *Learning, regret minimization and equilibria*. Cambridge University Press.

Blum, A., & Monsour, Y. (2007). From external to internal regret. *Journal of Machine Learning Research*, *8*, 1307–1324.

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

Bowling, M., & Veloso, M. (2002). Multiagent learning using a variable learning rate. *Artificial Intelligence*, *136*, 215–250.

Bowling, M. (2005). Convergence and no-regret in multiagent learning. *Advances in neural information processing systems*, *17*, 209–216.

Bowling, M., & Veloso, M. (2001). Rational and convergent learning in stochastic games. In *Proc. of the 17th Intl. Joint Conf. on Artificial Intelligence*, pp. 1021–1026.

Brown, G. W. (1951). Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, *13*(1), 374–376.

Busoniu, L., Babuska, R., & De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *38*(2), 156–172.

Busoniu, L., Babuska, R., De Schutter, B., & Ernst, D. (2010). *Reinforcement learning and dynamic programming using function approximators*. CRC Press.

Chalkiadakis, G., & Boutilier, C. (2003). Coordination in multiagent reinforcement learning: a Bayesian approach. In *Proc. of the 2nd intl. joint conf. on Autonomous Agents and MultiAgent Systems*, pp. 709–716. ACM.

Claes, D., Hennes, D., Tuyls, K., & Meeussen, W. (2012). Collision avoidance under bounded localization uncertainty. In *IROS*, pp. 1192–1198.

Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI/IAAI*, pp. 746–752.

Conitzer, V., & Sandholm, T. (2007). Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, *67*(1-2), 23–43.

Cressman, R. (2005). Stability of the replicator equation with continuous strategy space. *Mathematical Social Sciences*, *50*(2), 127–147.

Cross, J. G. (1973). A stochastic learning model of economic behavior. *The Quarterly Journal of Economics*, *87*(2), 239–266.

Dearden, R., Friedman, N., & Russell, S. (1998). Bayesian Q-learning. In *AAAI/IAAI*.

Ernst, D., Geurts, P., & Wehenkel, L. (2005). Tree-based batch mode reinforcement learning. In *Journal of Machine Learning Research*, pp. 503–556.

Fiorini, P., & Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, *17*, 760–772.

Fulda, N., & Ventura, D. (2007). Predicting and preventing coordination problems in cooperative Q-learning systems.. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-07)*, Vol. 2007, pp. 780–785.

Galstyan, A. (2013). Continuous strategy replicator dynamics for multi-agent q-learning. *Autonomous agents and multi-agent systems*, *26*(1), 37–53.

Gatti, N., Panozzo, F., & Restelli, M. (2013). Efficient evolutionary dynamics with extensive-form games. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, pp. 335–341.

Gibbons, R. (1992). *A Primer in Game Theory*. Pearson Education.

Gintis, H. (2009). *Game Theory Evolving* (2nd edition). University Press, Princeton NJ.

Gomes, E. R., & Kowalczyk, R. (2009). Dynamic analysis of multiagent Q-learning with ε-greedy exploration. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 369–376. ACM.

Hennes, D., Bloembergen, D., Kaisers, M., Tuyls, K., & Parsons, S. (2012). Evolutionary advantage of foresight in markets. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2012)*, pp. 943–950.

Hennes, D., Claes, D., & Tuyls, K. (2013). Evolutionary advantage of reciprocity in collision avoidance. In *Proc. of the AAMAS 2013 Workshop on Autonomous Robots and Multirobot Systems (ARMS 2013)*.

Hennes, D., Kaisers, M., & Tuyls, K. (2010). RESQ-learning in stochastic games. In *Adaptive and Learning Agents Workshop at AAMAS 2010*, p. 8.

Hennes, D., Tuyls, K., & Rauterberg, M. (2008). Formalizing multi-state learning dynamics. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pp. 266–272. IEEE Computer Society.

Hennes, D., Tuyls, K., & Rauterberg, M. (2009). State-coupled replicator dynamics. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 789–796. International Foundation for Autonomous Agents and Multiagent Systems.

Hofbauer, J., & Sigmund, K. (1998). *Evolutionary games and population dynamics*. Cambridge University Press.

Howell, M. N., Frost, G. P., Gordon, T. J., & Wu, Q. H. (1997). Continuous action reinforcement learning applied to vehicle suspension control. *Mechatronics*, *7*(3), 263–276.

Hu, J., & Wellman, M. P. (2003). Nash q-learning for general-sum stochastic games. *The Journal of Machine Learning Research*, *4*, 1039–1069.

Kaelbling, L., Littman, M., & Moore, A. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Kaisers, M., & Tuyls, K. (2010). Frequency adjusted multi-agent Q-learning. In *Proc. of 9th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pp. 309–315.

Kaisers, M. (2012). *Learning against Learning - Evolutionary Dynamics of Reinforcement Learning Algorithms in Strategic Interactions*. Ph.D. thesis, Maastricht University.

Kaisers, M., Bloembergen, D., & Tuyls, K. (2012). A common gradient in multi-agent reinforcement learning. In *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pp. 1393–1394.

Kaisers, M., & Tuyls, K. (2011). Faq-learning in matrix games: Demonstrating convergence near nash equilibria, and bifurcation of attractors in the battle of sexes. In *Workshop on Interactive Decision Theory and Game Theory (IDTGT 2011)*. Assoc. for the Advancement of Artif. Intel. (AAAI).

Kaisers, M., Tuyls, K., Parsons, S., & Thuijsman, F. (2009). An evolutionary model of multi-agent learning with a varying exploration rate. In *Proc. of The 8th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pp. 1255–1256. International Foundation for Autonomous Agents and Multiagent Systems.

Kapetanakis, S., & Kudenko, D. (2002). Reinforcement learning of coordination in cooperative multi-agent systems. *AAAI/IAAI*, *2002*, 326–331.

Kianercy, A., & Galstyan, A. (2012). Dynamics of boltzmann Q learning in two-player two-action games. *Phys. Rev. E*, *85*(4), 041145.

Klos, T., Van Ahee, G. J., & Tuyls, K. (2010). Evolutionary dynamics of regret minimization. In *Machine Learning and Knowledge Discovery in Databases*, pp. 82–96. Springer.

Lanctot, M. (2014). Further developments of extensive-form replicator dynamics using the sequence-form representation. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 1257–1264. International Foundation for Autonomous Agents and Multiagent Systems.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning.. In *ICML*, Vol. 94, pp. 157–163.

Maynard Smith, J., & Price, G. R. (1973). The logic of animal conflict. *Nature*, *246*(2), 15–18.

Mihaylov, M., Tuyls, K., & Nowé, A. (2014). A decentralized approach for convention emergence in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, *28*(5), 749–778.

Narendra, K. S., & Thathachar, M. A. L. (1974). Learning automata - a survey. *IEEE Transactions on Systems, Man, and Cybernetics*, *4*(4), 323–334.

Oechssler, J., & Riedel, F. (2001). Evolutionary dynamics on infinite strategy spaces. *Economic Theory*, *17*(1), 141–162.

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

Panait, L., & Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, *11*(3), 387–434.

Panozzo, F., Gatti, N., & Restelli, M. (2014). Evolutionary dynamics of Q-learning over the sequence form. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 2034–2040.

Phelps, S., Parsons, S., & McBurney, P. (2005). An evolutionary game-theoretic comparison of two double-auction market designs. In Faratin, P., & Rodrguez-Aguilar, J. (Eds.), *Agent-Mediated Electronic Commerce VI. Theories for and Engineering of Distributed Mechanisms and Systems*, Vol. 3435 of *Lecture Notes in Computer Science*, pp. 101–114. Springer Berlin Heidelberg.

Pipattanasomporn, M., Feroze, H., & Rahman, S. (2009). Multi-agent systems in a distributed smart grid: Design and implementation. In *Power Systems Conference and Exposition*, pp. 1–8. IEEE.

Ponsen, M., Tuyls, K., Kaisers, M., & Ramon, J. (2009). An evolutionary game-theoretic analysis of poker strategies. *Entertainment Computing*, *1*(1), 39–45.

Puterman, M. L. (1994). *Markov decision processes: Discrete dynamic stochastic programming*. John Wiley & Sons, New York.

Rodríguez, A., Vrancx, P., Grau, R., & Nowé, A. (2012). An RL approach to common-interest continuous action games. In *Proceedings of the 11th International Conference* *on Autonomous Agents and Multiagent Systems*, pp. 1401–1402. International Foundation for Autonomous Agents and Multiagent Systems.

Ruijgrok, M., & Ruijgrok, T. W. (2005). Replicator dynamics with mutations for games with a continuous strategy space. *arXiv preprint nlin/0505032*.

Santharam, G., Sastry, P. S., & Thathachar, M. A. L. (1994). Continuous action set learning automata for stochastic optimization. *Journal of the Franklin Institute*, *331*(5), 607–628.

Schaerf, A., Shoham, Y., & Tennenholtz, M. (1995). Adaptive load balancing: A study in multi-agent learning. *J. Artif. Intell. Res. (JAIR)*, *2*, 475–500.

Shoham, Y., Powers, R., & Grenager, T. (2007). If multi-agent learning is the answer, what is the question?. *Artificial Intelligence*, *171*(7), 365–377.

Singh, S., Kearns, M., & Mansour, Y. (2000). Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pp. 541–548. Morgan Kaufmann Publishers Inc.

Skyrms, B. (2004). *The stag hunt and the evolution of social structure*. Cambridge University Press.

Strens, M. (2000). A Bayesian framework for reinforcement learning. In *ICML*, pp. 943–950.

Sutton, R., & Barto, A. (1998). *Reinforcement Learning: An introduction*. MA: MIT Press, Cambridge.

't Hoen, P. J., Tuyls, K., Panait, L., Luke, S., & Poutré, J. A. L. (2005). An overview of cooperative and competitive multiagent learning. In *LAMAS*, pp. 1–46.

Tesauro, G. (2003). Extending q-learning to general adaptive multi-agent systems.. In *NIPS*, Vol. 4.

Thathachar, M., & Sastry, P. S. (2002). Varieties of learning automata: an overview. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *32*(6), 711–722.

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

Tuyls, K., Heytens, D., Nowé, A., & Manderick, B. (2003). Extended replicator dynamics as a key to reinforcement learning in multi-agent systems. In *Machine Learning: ECML 2003*, pp. 421–431. Springer.

Tuyls, K., & Nowé, A. (2005). Evolutionary game theory and multi-agent reinforcement learning. *The Knowledge Engineering Review*, *20*(01), 63–90.

Tuyls, K., & Parsons, S. (2007). What evolutionary game theory tells us about multiagent learning. *Artificial Intelligence*, *171*(7), 406–416.

Tuyls, K., Verbeeck, K., & Lenaerts, T. (2003). A selection-mutation model for q-learning in multi-agent systems. In *Proc. of 2nd Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2003)*, pp. 693–700, New York, NY, USA. ACM.

Tuyls, K., & Weiss, G. (2012). Multiagent learning: Basics, challenges, and prospects. *AI Magazine*, *33*(3), 41.

Tuyls, K., & Westra, R. (2009). Replicator dynamics in discrete and continuous strategy spaces. In Uhrmacher, A. M., & Weyns, D. (Eds.), *Multi-Agent Systems: Simulation and Applications*, pp. 215–240. CRC Press.

Van Kampen, N. (1992). *Stochastic Processes in Physics and Chemistry*. Elsevier Science Publishers, Amsterdam.

Verbeeck, K., Nowé, A., & Tuyls, K. (2005). Coordinated exploration in multi-agent reinforcement learning: an application to load-balancing. In *AAMAS*, pp. 1105–1106.

Von Neumann, J., & Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press.

Vrancx, P., Tuyls, K., Westra, R., & Nowé, A. (2008a). Switching dynamics of multi-agent learning. In *Proc. of the 7th intl. joint conf. on autonomous agents and multiagent systems (AAMAS 2008)*, pp. 307–313. International Foundation for Autonomous Agents and Multiagent Systems.

Vrancx, P., Verbeeck, K., & Nowé, A. (2008b). Decentralized learning in markov games. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *38*(4), 976–981.

Walsh, W., Das, R., Tesauro, G., & Kephart, J. (2002). Analyzing complex strategic interactions in multi-agent systems. In *AAAI-02 Workshop on Game-Theoretic and Decision-Theoretic Agents*.

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3), 279–292.

Weibull, J. W. (1997). *Evolutionary game theory*. MIT press.

Wheeler Jr, R., & Narendra, K. S. (1986). Decentralized learning in finite markov chains. *IEEE Transactions on Automatic Control*, *31*(6), 519–526.

Whiteson, S., & Stone, P. (2006). Evolutionary function approximation for reinforcement learning. *The Journal of Machine Learning Research*, *7*, 877–917.

Wiegand, R. P. (2003). *An Analysis of Cooperative Coevolutionary Algorithms*. Ph.D. thesis, George Mason University.

Wunder, M., Littman, M. L., & Babes, M. (2010). Classes of multiagent q-learning dynamics with epsilon-greedy exploration. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 1167–1174.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proc. of the 20th Intl. Conf. on Machine Learning (ICML-2003)*.