# Replicator Dynamics for Multi-agent Learning An Orthogonal Approach

Michael Kaisers
*Maastricht University, P.O. Box 616, 6200 MD Maastricht*

August 28, 2009

**Abstract**

Today's society is largely connected and many real life applications lend themselves to be modeled as multi-agent systems. Although such systems as well as their models are desirable, e.g. for reasons of stability or parallelism, they are highly complex and therefore difficult to understand or predict. Multi-agent learning has been acknowledged to be indispensable to control or find solutions for such systems. Recently, evolutionary game theory has been linked to multi-agent reinforcement learning. However, gaining insight into the dynamics of games, especially if time dependent, remains a challenging problem. This article introduces a new perspective on the reinforcement learning process described by the replicator dynamics, providing a tool to design time dependent parameters of the game or the learning process. This perspective is orthogonal to the common view of policy trajectories driven by the replicator dynamics. Rather than letting the time dimension collapse, the set of initial policies is considered to be a particle cloud that approximates a distribution and we look at the evolution of this distribution over time. First, the methodology is described, then it is applied to an example game and viable extensions are discussed.

**Keywords:** Reinforcement learning, Evolutionary game theory

## 1 Introduction

The world of today is full of networks and connected systems. As a consequence, the assumption that a system runs in actual isolation of any other actor does not withstand reality. Hence, many domains need to be modeled as multi-agent systems in order to account for their inherent complexity. However, the models yield a complexity that makes them hard to understand, predict or control. As this is realized, multi-agent learning gains popularity to find solutions to or control these systems [7, 8].

Learning in multi-agent environments is significantly more complex than single-agent learning as the dynamics to learn change by the learning processes of other agents. This makes predicting learning behavior of learning algorithms in multi-agent environments difficult. They are not only situated in a non-stationary environment but also need to deal with incomplete information and communication limits. In non-stationary environments the Markov property does not hold which makes all proofs of convergence to optimal policies from single-agent learning that are based on that assumption inapplicable. This reduces the theoretical framework available for multi-agent learning. More recently, evolutionary game theory with less strong assumptions than classical game theory could be linked to reinforcement learning and provides useful insights into the learning dynamics [1, 3, 11].

The learning dynamics are commonly visualized by showing the directional field plot of the replicator dynamics or showing policy trajectories with the time dimension collapsed into a surface. Both views work well for dynamics that do not change over time but provide little guidance when the game or the learning algorithm uses a parameter that is time dependent. In particular, the directional field plot can only capture the dynamics at one point in time. Hence, several independent plots are needed for changing dynamics and a gap remains in the transition between them. The trajectory view becomes unclear when circles occur or the dynamics change, in which case lines may intersect and clutter the plot. Furthermore, reducing the time dimension into a flat surface hinders the interpretation of time dependent artifacts. In addition, the higher the resolution (the more trajectories are plotted), the more crowded the plot and the harder it becomes to

interpret. As a result, parameter tuning is a cumbersome task that often results in ad hoc trial and error approaches.

In order to tackle these problems, this article will answer the question how to extract more information from dynamical systems, especially for time dependent dynamics, with the goal of facilitating the systematical design of time dependent parameters. This is achieved by taking on a new perspective that is orthogonal to the common view of policy trajectories.

The remainder of this article is structured as follows: Section 2 introduces relevant concepts from game theory and reinforcement learning, and Section 3 proposes a new perspective on the process driven by the replicator dynamics. Section 4 demonstrates the new methodology on an example game and Section 5 concludes the paper with a discussion of viable extensions.

# 2 Background

## 2.1 Game theory

Classical game theory is the mathematical study of strategic conflicts of rational agents. The central concept is the *game*, which comprises a set of players $I = \{1, 2, \ldots, n\}$ and a set of available pure strategies $S_i = \{1, 2, \ldots, k_i\}$ for each player $i$, for $n$ and $k_i$ some finite integer. For a more general introduction we refer the interested reader to [3, 12].

The players of normal form games are assumed to choose their pure strategies simultaneously and independently and receive a payoff that is dependent on the joint strategy profile $s \in S_1 \times \ldots \times S_n$. The payoff for two-player normal form games can be described by two matrices $A$ and $B$, where for any joint strategy $(i, j)$, $A_{ij}$ denotes the payoff to player one and $B_{ij}$ describes the payoff to player two.

As we are only considering repeated stateless games, the policy of each player can be described by a probability distribution over the available actions at each point in time $t$. The two-player game in this example will use the notation $x$ and $y$ for the policy vectors of the first and second player respectively.

## 2.2 Reinforcement learning

Reinforcement Learning (RL) has originally been studied in the context of single-agent environments. An agent receives a numerical reward signal, which it seeks to maximize. The environment provides this signal as a feedback on the sequence of actions that has been executed by the agent. Learners relate the reward signal to previously executed actions to learn a policy that maximizes the expected future reward [9].

The environment is defined by the normal form game and the reinforcement learner updates the policy. A very simple policy iterator is the *Cross Learning* algorithm, a specific type of learning automata. When action $i$ is selected and reward $r(t) \in [0, 1]$ is received at time $t$, then policy $x$ is updated according to the following equation:

$$x_i(t+1) \leftarrow (1 - r(t))x_i(t) + r(t)$$
$$x_j(t+1) \leftarrow (1 - r(t))x_j(t), \text{ for all } j \neq i$$

The policy change induced by this learner has been shown to converge under infinitesimal time steps to the replicator dynamics [1]. For the sake of clarity, this model is used for the further study in this article. However, it is worth mentioning that other learning algorithms can be described by similar differential equations, e.g. Q-learning with a Boltzmann exploration scheme has been shown to converge to an extension of the replicator dynamics [11]. The evolutionary description of reinforcement learning is detailed in the following subsections using the example of Cross learning.

## 2.3 Evolutionary game theory

Evolutionary game theory takes a rather descriptive perspective, replacing hyper-rationality from classical game theory by the concept of natural selection from biology [6]. The two central concepts of evolutionary game theory are the replicator dynamics and evolutionary stable strategies. The replicator dynamics presented in the next subsection describe the evolutionary change in the population. They are a set of differential equations that are derived from biological operators such as selection, mutation and cross-over. The evolutionary stable strategy describes the resulting asymptotic behavior of this population. However, their examination is beyond the scope of this article. For a detailed discussion, we refer the interested reader to [4, 5].

### 2.3.1 Replicator dynamics

The replicator dynamics from evolutionary game theory formally define the population change over time. A population comprises a set of individuals, where the species that an individual can belong to represent the pure strategies. The utility function can be interpreted as the Darwinian fitness of each species. The distribution of the individuals on the different strategies can be described by a probability vector that is equivalent to a policy. Hence, there is a second view on the evolutionary process: The population may also represent competing strategies within the mind of one agent, who learns which one to apply. The evolutionary pressure by natural selection can be modeled by the replicator equations. They assume this population to evolve such that successful strategies with higher payoffs grow while less successful ones decay. These dynamics are formally connected to reinforcement learning [1, 10, 11]. Assume a two-player normal form game with payoff matrices $A$ and $B$ for player one and two respectively, and let the policies of player one and two be represented by the probability vectors $x = (x_1, \ldots, x_k)$ and $y = (y_1, \ldots, y_k)$, where $x_i$ and $y_i$ indicate the probability to play action $i$. The two-player replicator dynamics that relate to the learning process of *Cross Learning*, a simple learning automaton, are given by the following set of differential equations, where $e_i$ is the $i^{\text{th}}$ unit vector:

$$\begin{aligned} \dot{x}_i &= x_i \left[ e_i A y^T - x A y^T \right] \\ \dot{y}_i &= y_i \left[ x B e_i^T - x B y^T \right] \end{aligned} \tag{1}$$

The change in the fraction playing action $i$ is proportional to the difference between the expected payoffs $e_i A y$ and $x B e_i$ of action $i$ against the mixing opponent, and the expected payoff $xAy$ and $xBy$ of the mixed strategy $x$ against the mixed strategy $y$. Hence, above average actions strive while below average actions decay. The replicator dynamics maintain the probability distribution, thus $\sum_i \dot{x}_i = 0$. The examples used in this article are constraint to two actions. which implies $\dot{x}_1 = -\dot{x}_2$ and $\dot{y}_1 = -\dot{y}_2$.

### 2.3.2 Policy trajectories

The replicator dynamics describe how the policy changes over time, dependent on the game and the policy itself. Starting with a set of policies, we can follow this change over a period of time. The path that is taken is referred to as the policy trajectory. In a game with $k$ actions for each player, the policy trajectory of $p$ players can be specified with $(k-1)^p + 1$ dimensions. In the case of two-player two-action games, the unit square yields the policy space, as we can specify $x_1$ as a full characterization of $x = (x_1, 1 - x_1)$, and $y_1$ similarly. One more dimension is optionally needed for an exploded view showing the time dimension.

# 3 Method

This section shows the learning process in a new perspective which is orthogonal to viewing policy trajectories in the classical way. Figure 1 shows 20 trajectories in an expanded view of policy space against time. Instead of looking at it from the top down, we suggest cutting slices at different points in time and looking at the distribution of trajectory points where they intersect these slices.

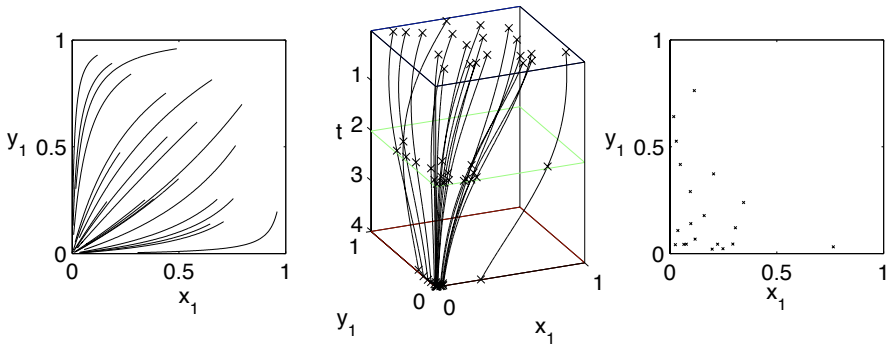


**Figure 1:** An expanded view of 20 policy trajectories (middle), the common perspective collapsing the time dimension (left), showing the trajectories as a flat image, and the proposed orthogonal perspective (right), looking at the second slice that intersects the trajectories at the indicated points.

The idea behind considering distributions rather than single trajectories is to obtain a more holistic view of the learning process. In the end, learning is a homeomorphic time dependent transformation of the policy space. As such, we can look at its influence on the whole space, e.g. by looking at the spacing between the trajectories, rather than only looking at individual policy trajectories. In order to do so, a set of particles is drawn from an initial distribution, in the given examples a uniform distribution, and subjected to a velocity field defined by the replicator dynamics. As time evolves, the distribution is transformed and the density of the particles changes. This allows to make statements of the following kind: Assuming any policy was initially equally likely and these policies evolve according to the replicator dynamics, then after time $t$ has passed, $p$ percent of the policies have converged to attractor $a$ with at most distance $\epsilon$.

After some time, the simulation can be stopped and labels can be applied according to the eventual distribution. A certain percentage of particles can be considered converged to some attractors, assuming they are in the neighborhood of a stable point and that point is attracting in that neighborhood. Other particles can be labeled as not converged or the similar. Finally, these labels can be applied to earlier slices including the initial slice, revealing the basins of attraction. Although these basins can also be read from the directional field plot of the replicator dynamics, this approach is more general as it can be applied to dynamics that are controlled by a time dependent parameter.

In addition, this allows to judge the convergence of a fraction of the policy space that is bound by a surface by considering the velocity field only on that surface. Due to the fact that the dynamics describe a continuous process and the transformation by the replicator dynamics is thereby a homeomorphism, everything that is added or substracted from the trapped percentage has to go through the surface. This is related to the idea of *divergence* from physics [2]. It allows to focus attention on the surface that may be just a small subspace of the whole policy space, e.g. a hypersphere with radius $\epsilon$ around an attractor. In many cases, the velocity field in this small neighborhood can be guaranteed to be rather static although the dynamics of other areas of the policy space may change quite substantially.

It is common to assume every policy initially equally likely, i.e. applying an initially uniform distribution. However, this approach allows to use an arbitrary initial distribution which can be used to model specific knowledge about the initial learning behavior. Furthermore, the policy distribution can also be generated from Q-value distributions, in case a Q-learning algorithm should be modeled. Using a similar evolution as the replicator dynamics in the Q-value space, the distribution can be evolved which allows comparing Boltzmann exploration to other exploration schemes that do not have a bijective action selection function[1] and can therefore not be solely described by dynamics in the policy space.

# 4 Experiments

This section demonstrates the proposed methodology on an example game that is controlled by a parameter that may change its value at one point in time. The game describes the following situation:

> "There are two new standards that enable communication via different protocols. The consumers and suppliers can be described by probability vectors that show which standard is supported by which fractions. One protocol is $20\%$ more energy efficient, hence the government wants to support that standard. Usually, the profit of the consumers and suppliers are directly proportional to the fraction of the opposite type that supports their standard. However, the government decides to subsidize early adopters of the better protocol.
> Such subsidies are expensive and the government only wants to spend as much as necessary. They have no market research information and consider any distribution of supporters on both sides equally likely. Furthermore, they know that the supporters are rational and their fractions will change according to the replicator dynamics. The question is, how long is the subsidy necessary to guarantee that the better standard is adopted in $95\%$ of the possible initial policies."

This is a variation of the pure coordination game. A subsidy parameter $s \in \{0, 11\}$ is added, which can be used to make one action dominant. As a result, coordination on the Pareto optimal equilibrium is facilitated. Figure 2 displays the payoff bi-matrix numerically.

[1] Strictly speaking, Boltzmann action selection is also not a bijection, as it leaves one degree of freedom when computing Q-values from policies. However, each policy change relates to a Q-value change and vice versa, which is not the case in other exploration schemes such as epsilon-greedy.

| | $S_1$ | $S_2$ |
|---|---|---|
| $S_1$ | $10, 10$ | $0, s$ |
| $S_2$ | $s, 0$ | $12, 12$ |

| | $S_1$ | $S_2$ |
|---|---|---|
| $S_1$ | $10, 10$ | $0, 0$ |
| $S_2$ | $0, 0$ | $12, 12$ |

| | $S_1$ | $S_2$ |
|---|---|---|
| $S_1$ | $10, 10$ | $0, 11$ |
| $S_2$ | $11, 0$ | $12, 12$ |

**Figure 2:** The payoff bi-matrix for the subsidy game (left) and its realizations for $s = 0$ (middle) and $s = 11$ (right). Player one chooses a row, player two chooses a column. The first number of the selected action combination represents the payoff to player one and the second number the payoff to player two.

The dynamics of the game can be visualized by showing the directional field plot of the replicator dynamics as shown in Figure 3. It can be observed that a large fraction of the policy space would converge to the suboptimal standard in the unsubsidized game, while all policies would converge to the optimum in the subsidized game. However, it is impossible to infer the correct time to switch between the two games.

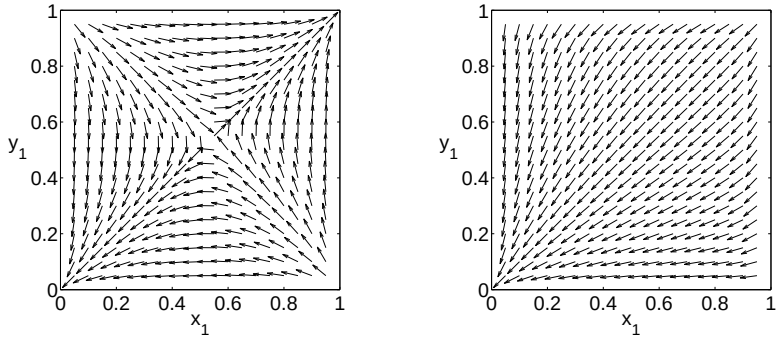


**Figure 3:** The dynamics of the game without subsidy (left) and with subsidy (right).

The second classical way to look at the dynamics are policy trajectories. These will follow the directional change and are depicted in Figure 4. Similar to the replicator dynamics, this view neatly explains the dynamics of the individual parts of the game, but does not allow to infer the right time to switch from the one to the other.

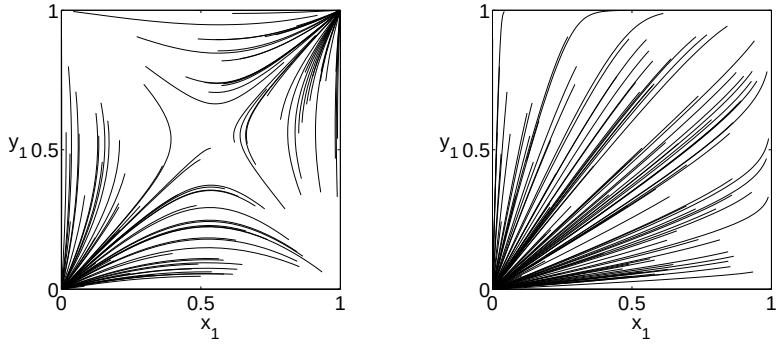


**Figure 4:** The trajectories in the game without subsidy (left) and with subsidy (right).

Another possible approach is the visualization of trajectories with transitions from one game to the other at different points in time. Figure 5 shows the trajectories of the subsidy game when transition from $s = 11$ to $s = 0$ takes place at $t = \{0.1, 0.3, 0.5\}$. Although it can be observed that less trajectories converge suboptimal the later the switch occurs, this approach requires to guess the right time of transition. Furthermore, the view is cluttered by intersecting lines and readability does not allow to increase the number of trajectories.

In order to obtain insight into the time dependent artifacts of these dynamics, the new perspective will be applied. Answering the question when to switch requires 2 steps:

- Find the basin of the unsubsidized game
- Determine the time when the subsidized dynamics have driven $95\%$ of the initial policies into the basin of the global optimum in the unsubsidized game.

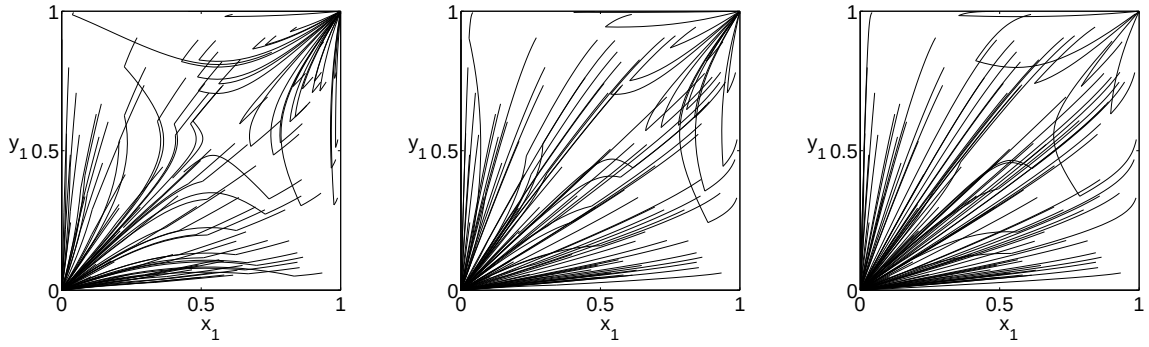


**Figure 5:** The trajectory plot for the subsidy game with transition from the subsidized to the unsubsidized game at $t = \{0.1, 0.3, 0.5\}$ (left to right).

Step one is shown in Figure 6. Particles are drawn from a uniform initial distribution and evolved according to the replicator dynamics. After $t = 1.2$ the particles are considered converged and receive a label (a dot for the optimum and an x for the suboptimal attractor). Subsequently, the label is applied to all slices before plotting. From the labels on the initial slice, the basin boundary is deduced, which is marked by the dashed line.

In step 2, shown in Figure 7, the boundary that has been inferred from step one is used to monitor the percentage of the initial policy space that would converge to the optimum if the game was switched at that time instance. The simulation advances until the subsidized dynamics have pushed $95\%$ of the initial policies into the basin of attraction of the global optimum in the unsubsidized game. Then, the game is switched and the simulation shows convergence to the according attractors. Repeating the experiment $n = 1000$ times, we find that the time to bring $95\%$ to the basin is $0.495 \pm 0.0357$ (indicating one standard deviation). A histogram of the distribution is given in Figure 8.

This section has demonstrated the advantages of the proposed perspective. It has been shown that the new methodology allows the systematic study and design of time dependent parameters in order to achieve desired effects, in the example a specific convergence behavior. The next section concludes the article with a discussion of contributions and viable extensions to the introduced approach.

# 5 Discussion and conclusions

The contributions of this article can be summarized as follows: A new perspective on dynamical systems driven by time dependent replicator dynamics has been proposed. An illustrative example of a two-agent two-action game has been discussed, and the method has been shown to naturally reveal time dependent properties of the system. This facilitates designing parameters with a systematic approach rather than setting them ad hoc. While a rather simple example game was studied for the sake of clarity, the approach is

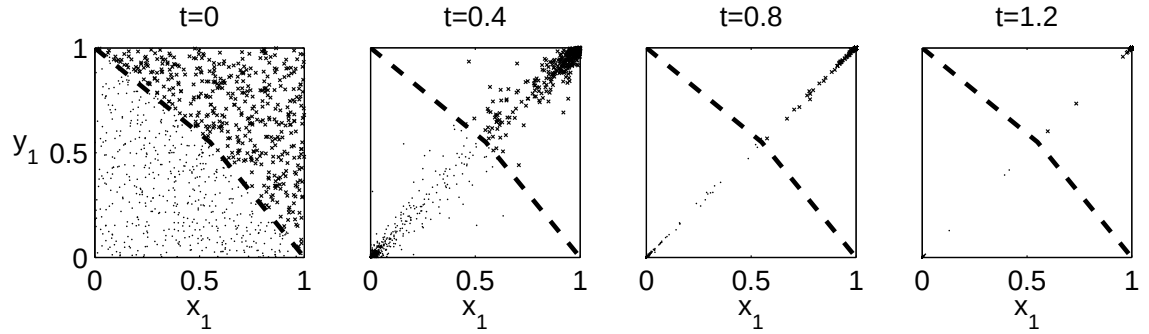


**Figure 6:** This figure shows the evolution of particles drawn from a uniform initial distribution, revealing the basins of attraction of the unsubsidized game. Labels are applied according to the last slice and the dashed line is inferred from the labels to be the boundary between the basins of attraction.

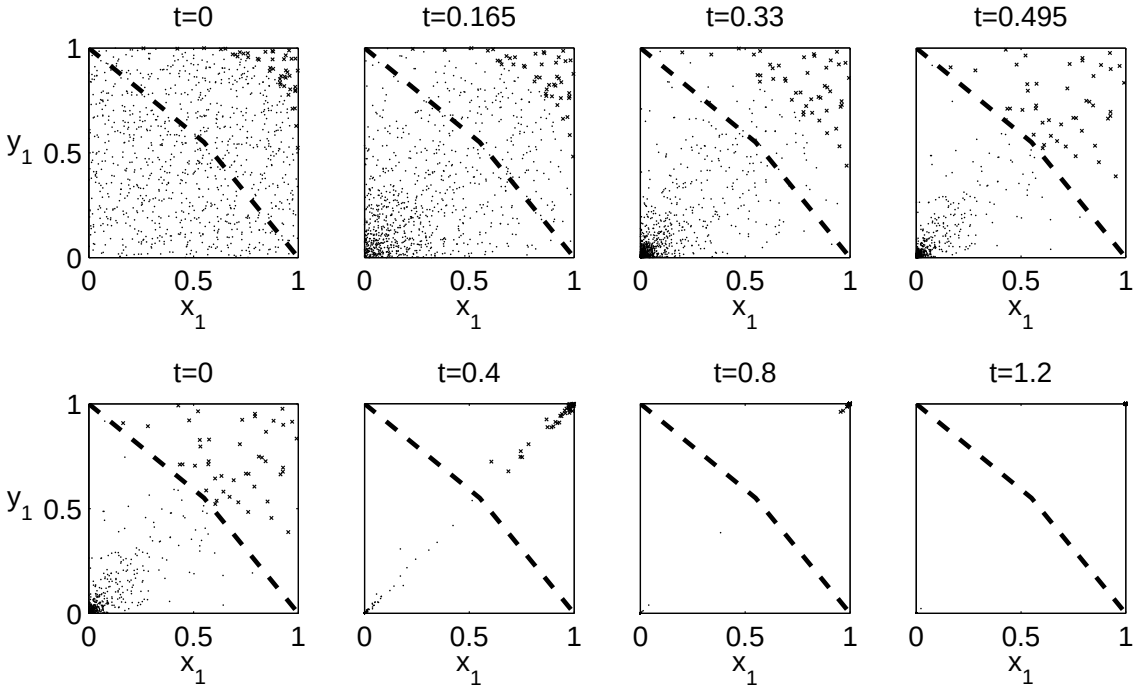


**Figure 7:** The top row shows the evolution under the subsidized game until $95\%$ of the policy space are in the basin for the global optimum of the unsubsidized game. The lower row shows the further evolution in the unsubsidized game.

general in the number of actions and can be applied to arbitrary initial distributions. In addition, it naturally generalizes to any number of agents when the reward matrix is considered to be a reward function on two strategy variables.

The parameter design methodology can be transfered to other parameters that change the replicator dynamics, most prominently the temperature function for Q-learning with a Boltzmann exploration scheme. Choosing an appropriate temperature function has long been a heuristic search and can now be tackled systematically to achieve a desired convergence distribution.

Furthermore, the ideas presented in this paper have the strong potential to be further developed. The current approach can be seen as a particle simulation, where the replicator dynamics determine the velocity field that describes the movement of each particle, and the particle density describes a probability distribution. The authors have taken preliminary steps to make the transition to describe this probability density function as a continuous function, deriving the density change directly from the replicator dynamics. This will remove the stochasticity introduced by approximating the probability density by quantized particles.

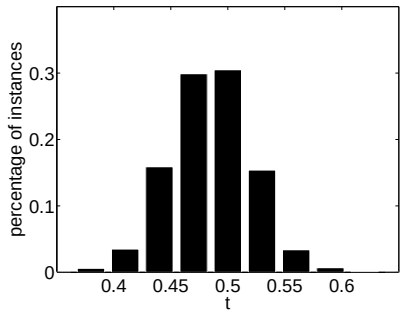


**Figure 8:** Histogram of times at which the velocity field of the subsidized game has driven $95\%$ of the particles into the basin of attraction of the global optimum in the unsubsidized game. The sample size is $n = 1000$, with a mean of $0.495$ and a standard deviation of $0.0357$.

Another viable extension considers a distribution of Q-values and the distribution's evolution, deriving the according policy distributions from it. This enables the comparison of exploration schemes such as Boltzmann and epsilon-greedy exploration. Finally, this model is extendable to multiple states and continuous strategy spaces, which will compliment the theoretical framework for multi-agent learning.

# 6 Acknowledgements

This research was partially sponsored by a TopTalent2008 grant of the Netherlands Organisation for Scientific Research (NWO).

# References

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

[2] Richard P. Feynman, Robert B. Leighton, and Matthew Sands. *The Feynman Lectures on Physics including Feynman's Tips on Physics: The Definitive and Extended Edition*. Addison Wesley, August 2005.

[3] C. M. Gintis. *Game Theory Evolving*. University Press, Princeton, June 2000.

[4] Morris W. Hirsch, Stephen Smale, and Robert Devaney. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Academic Press, 2002.

[5] Josef Hofbauer and Karl Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, 2002.

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

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

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

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

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

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

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