Analyzing Multi-agent Reinforcement Learning Using Evolutionary Dynamics

Pieter Jan 't Hoen[1] and Karl Tuyls[2,⋆]

[1] Center for Mathematics and Computer Science (CWI)
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
[2] Computational Modeling Lab, Department of Computer Science
Vrije Universiteit Brussel, Belgium
`hoen@cwi.nl, ktuyls@vub.ac.be`

Abstract. In this paper, we show how the dynamics of Q-learning can be visualized and analyzed from a perspective of Evolutionary Dynamics (ED). More specifically, we show how ED can be used as a model for Q-learning in stochastic games. Analysis of the evolutionary stable strategies and attractors of the derived ED from the Reinforcement Learning (RL) application then predict the desired parameters for RL in Multi-Agent Systems (MASs) to achieve Nash equilibriums with high utility. Secondly, we show how the derived fine tuning of parameter settings from the ED can support application of the COllective INtelligence (COIN) framework. COIN is a proved engineering approach for learning of cooperative tasks in MASs. We show that the derived link between ED and RL predicts performance of the COIN framework and visualizes the incentives provided in COIN toward cooperative behavior.

1 Introduction

In this work, we present a novel approach to study Reinforcement Learning (RL) applied to Multi-Agent Systems (MASs), a growing area of research [3, 10, 7, 1, 9]. The challenge for a MAS is to achieve a high degree of performance as the agents learn in parallel. In this work, the dynamics of the RL process are visualized and analyzed from a perspective of Evolutionary Dynamics. More specifically, we show how ED from Evolutionary Game Theory (EGT) can be used as a model for Q-learning in stochastic games. We show a one-to-one connection between RL and the ED from evolutionary game theory. More precisely, we connect the Replicator Dynamics (RD) and a mutation term mathematically with Q-learning. Analysis of the evolutionary stable strategies and attractors of the ED aid in setting the parameters for the RL algorithms to achieve desired Nash equilibriums with high utility. This is a step forward from the classic approach of trial and error when deciding upon what parameters to use in MAS RL. The link between RL and ED can help in understanding the development

---

of the MAS during learning. We show how the ED can visualize some of the phenomena troubling MAS learning and assist in analysis of the performance and convergence properties of the MAS.

We illustrate our contributions using Dispersion Games [6]. In this cooperative game, n agents each have to decide which of the k tasks they are to undertake. Agents acting in parallel and using local feedback with no central control must learn to arrive at an optimal distribution over the available tasks. Such problems are typical for a growing class of large-scale distributed applications such as load balancing, niche selection, division of roles within robotics, or application in logistics. We investigate the case for when $n = k$ which we call *full Dispersion Games*. Full utility is achieved only when every one of the n tasks is chosen by exactly one of the n agents. The joint action state space is n^n while there are only $n!$ desirable Nash Equilibriums with full utility. This makes the full Dispersion Game a challenging task for growing n and a possible benchmark for MAS applications.

We present the ED of this full dispersion game for n≥2 agents and show the match between the predicted behavior from EGT perspective and the observed behavior in the MAS using RL. Strong attractors in the ED become strategies to which agents will converge with high probability if seen from a RL (Q-learner) perspective. Furthermore, the settings in the ED for desirable attractors indicate the correct choice for parameters in the used RL algorithms.

As the second part of our contribution, we present work on the Replicator Dynamics applied to the COllective INtelligence (COIN) framework. COIN is used in cooperative MASs to induce incentives for actions by agents that promote the global utility of the system. This is in line with a growing collection of work where the goal is to establish conditions for MASs such that they are most likely to exhibit good emergent behavior [2, 11, 3]. The effectiveness of the COIN top-down approach has been demonstrated by applying the COIN framework to a number of example problems: network routing [25], increasingly difficult versions of the El Farol Bar problem [18], Braess' paradox [19], and complex token retrieval tasks [17]. Analyzing the COIN approach from the ED perspective is shown to give an indication through the attractors for ED as to what parameters to apply for COIN. Furthermore, the ED visualize the incentives in COIN for agents to strive for a high global utility using local learning. The ED hence offer the possibility to Analise advanced MAS design methodologies.

The rest of this document is structured as follows. Section 2 introduces the ED and the link with RL. Section 3 shows the analysis of the full Dispersion Games. Section 4 presents the COIN framework and Section 5 presents the use of the ED in COIN. Section 6 discusses and concludes.

2   Replicator Dynamics and RL

This section provides some background in learning in games, RL, and the replicator dynamics. Game theory [12, 13, 8, 5, 23], offers the mathematical foundation for the analysis and synthesis of problems in decentralized control. A game

consists of more than one player and results in an outcome for every player depending on the overall behavior of all the players. Formally it can be described by a tuple $(n, A_1 \ldots A_n, R_1 \ldots R_n)$ where n is the number of players, A_i the set of actions available to player i and $R_i : A_1 \times \ldots \times A_n \to \Re$ is the payoff function for player i. When games are played repeatedly a sequential decision problem arises. In this paper players are modeled as Q-learners [22], which are described in the following subsection.

2.1   Q-Learning in Games

A model of Reinforcement Learning consists of: a discrete set of environment states, a discrete set of agent actions and a set of scalar reward signals. On each step of interaction the agent receives a reward and some indication of the current state of the environment, and chooses an action. The agent's job is to find a policy, i.e. a mapping from states to actions, that maximizes some long-term measure of reward.

Common Reinforcement Learning methods, which can be found in [16] are structured around estimating value functions. A value of a state or state-action pair, is the total amount of reward an agent can expect to accumulate over the future, starting from that state. Q-learning, investigated here, is an adaptive value iteration method (see [16, 22]), which bootstraps its estimate for the state-action value $Q_{t+1}(s, a)$ at time $t + 1$ upon its estimate for $Q_t(s', a')$ with s' the state where the learner arrives after taking action a in state s:

$$Q_{t+1}(s, a) \leftarrow (1 - \alpha)Q_t(s, a) + \alpha(r + \gamma \, max_{a'} Q_t(s', a')) \tag{1}$$

with α the usual step size parameter, γ a discount factor, and r the immediate reinforcement.

2.2   The Replicator Equations

The basic concepts and techniques developed in EGT were initially formulated in the context of evolutionary biology [23, 13]. One way in which EGT proceeds is by constructing a dynamic process in which the proportions of various strategies in a population evolve. Examining the expected value of this process gives an approximation which is called the Replicator Dynamics (RD). RD highlights the role of selection; it describes how systems consisting of different strategies change over time. They are formalized as a system of differential equations. Each replicator (or genotype) represents one (pure) strategy. This strategy is inherited by all the offspring of the replicator. The general form of a replicator dynamic is the following:

$$\frac{dx_i}{dt} = [(A\mathbf{x})_i - \mathbf{x} \cdot A\mathbf{x}]x_i \tag{2}$$

In equation (2), x_i represents the density of strategy i in the population, A is the payoff matrix which describes the different payoff values each individual replicator receives when interacting with other replicators in the population.

The state of the population ($\mathbf{x}$) can be described as a probability vector $\mathbf{x} = (x_1, x_2, ..., x_J)$ which expresses the different densities of all the different types of replicators in the population. Hence $(A\mathbf{x})_i$ is the payoff which replicator i receives in a population with state x and $\mathbf{x} \cdot A\mathbf{x}$ describes the average payoff in the population. The growth rate $\frac{\frac{dx_i}{dt}}{x_i}$ of the population share using strategy i equals the difference between the strategy's current payoff and the average payoff in the population. For further information we refer the reader to [23, 8].

For the case of 2 agents p and q, we need two systems of differential equations: one for the player (p) and one for the player (q). In this case p_j represents the density of strategy j in the strategy population of the row player and q_i the density of strategy i in the strategy population of the column player. This setup corresponds to a RD for asymmetric games. If $A = B^t$, equation (2) would again emerge. This translates into the following replicator equations for the two populations:

$$\frac{dp_i}{dt} = [(A\mathbf{q})_i - \mathbf{p} \cdot A\mathbf{q}]p_i, \frac{dq_i}{dt} = [(B\mathbf{p})_i - \mathbf{q} \cdot B\mathbf{p}]q_i \qquad (3)$$

As can be seen in equations 3, the growth rate of the types in each population is now determined by the composition of the other population. Note that, when calculating the rate of change using these systems of differential equations, two different payoff matrices (A and B) are used for the two different players.

2.3   The Q-Learning Dynamics

In this section we briefly repeat the main results of [21]. In this paper however, we will extend the results of that previous work. More precisely, we will apply these results to stochastic Dispersion Games. The experiments of [21] were conducted in one-stage games. In this paper we will extend this approach to multi-state one-stage games, i.e. stochastic Dispersion Games. Moreover, we will show that this evolutionary game theoretic approach enhances the COIN framework of Wolpert et al. and we will show that this approach is scalable to Dispersion Games with many agents.

We now first present the relation between Q-learning and the RD (see sections 2.1 and 2.2)[1]. More precisely we present a continuous time limit of the Q-learning model, where Q-values are interpreted as Boltzmann probabilities for the action selection.

For simplicity we discuss results as games between 2 players, and hence also two tasks. Note that this scenario is also interpreted as a game of one agent versus the rest of the MAS. Scenarios (not shown here) of the row player opposing a column player representing one or more agents in the MAS targeting the same tasks as the row agent were illustrative of behavior of the row and the collective of the column agent in such settings. A good understanding of the MAS was found by analyzing the behavior of one row player and his preference for one

---

[1] The reader who is interested in the complete derivation and discussion is referred to [21, 20].

task compared to the other tasks while playing versus respectively one, two, or more opposing agents playing the role of the column player.

Each agent(or player(s)) has a probability vector over his action set, more precisely $x_1, ..., x_n$ over action set $a_1, ..., a_n$ for the first player and $y_1, ..., y_m$ over $b_1, ..., b_m$ for the second player. Formally the Boltzmann distribution is described by,

$$x_i(k) = \frac{e^{\tau Q_{a_i}(k)}}{\sum_{j=1}^{n} e^{\tau Q_{a_j}(k)}} \tag{4}$$

where $x_i(k)$ is the probability of playing strategy i at time step k and τ is the temperature. The temperature[2] determines the degree of exploring different strategies. As the trade-off between exploration-exploitation is very important in RL, it is important to set this parameter correctly. Now suppose that we have payoff matrices A and B for the 2 players. Calculating the time limit, as established in [20], results in,

$$\frac{dx_i}{dt} = x_i \alpha \tau ((A\mathbf{y})_i - \mathbf{x} \cdot A\mathbf{y}) + x_i \alpha \sum_j x_j ln(\frac{x_j}{x_i}) \tag{5}$$

for the first player and analogously for the second player in,

$$\frac{dy_i}{dt} = y_i \alpha \tau ((B\mathbf{x})_i - \mathbf{y} \cdot B\mathbf{x}) + y_i \alpha \sum_j y_j ln(\frac{y_j}{y_i}) \tag{6}$$

Comparing (5) or (6) with the RD in (2), we see that the first term of (5) or (6) is exactly the RD and thus takes care of the selection mechanism, see [23]. The mutation mechanism for Q-learning is therefore left in the second term, and can be rewritten as:

$$x_i \alpha \sum_j x_j ln(x_j) - ln(x_i) \tag{7}$$

In equation (7) we recognize 2 entropy terms, one over the entire probability distribution x, and one over strategy x_i.

Relating entropy and mutation is not new. It is a well known fact [14, 15] that mutation increases entropy. In [15], it is stated that the concepts are familiar with thermodynamics in the following sense: the selection mechanism is analogous to *energy* and mutation to *entropy*. So generally speaking, mutations tend to increase entropy. Exploration can be considered as the mutation concept, as both concepts take care of providing variety.

Equations 5 and 6 now express the dynamics of both Q-learners in terms of Boltzmann probabilities, from which the RD emerge. In the next section we apply the equations to Dispersion Games.

---

[2] Used in the literature either in the numerator or in the denominator in Equation 4.

3   Applying the Q-Learning Dynamics to Stochastic Games

In this section we apply the Q-learning dynamics to multi-state stochastic games. In table 1a, a stochastic game is defined. The payoffs are listed as tuples for agents one and two respectively. This is the Dispersion Game for two agents where one agent is called the row player and the other the column player. The $i-th$ row for a row player represents the choice for execution of task t_i, and similarly the $i-th$ column for the column player . The highest possible global utility is achieved by the two players if (and only if) different tasks are chosen. Otherwise, the overly chosen task is randomly assigned (with 50% (0.5) probability) with reward 1 to either the row or column player and 0 for the other player. In this case, only half of the full global utility is achieved in the Dispersion Game.

Table 1. Example of a stochastic (multi-state) game

	(a)		(b)				(c)	

(a)

0.5 1,0	
0.5 0,1	1,1
1,1	**0.5** 1,0
	0.5 0,1

(b)

state 1		state 2		state 3		state 4	
1,0	1,1	0,1	1,1	0,1	1,1	1,0	1,1
1,1	1,0	1,1	1,0	1,1	0,1	1,1	0,1

(c)

0.5,0.5	1,1
1,1	0.5,0.5

The stochastic payoff table 1a has four possible states that each represent one of the possible outcomes. These are expanded in payoff Table 1b.

For each of these different states we can now apply the ED equations 5 and 6 by filling in the payoff tables for A and B for the row and column player. Doing this allows us to plot the direction field for each state. This is a general approach to reduce stochastic payoffs to ED we can plot as in [21].

We show two direction plots in Figures 1 for a temperatures of $\tau = 2$ ((a) and (b)), and for $\tau = 10$ ((c) and (d)). The plots show the preferences of the choice of strategies for the row and column players for the representative states one and two of Table 1b respectively. The direction fields of the other two states (not shown) are the complement of the fields for state 1 and 2. The preference the row player has for choosing task t_1 is shown along the x axis and the preference the column player has for the same task is shown along the y axis. A point (p_1, p_2) in a graph is hence the joint (but independent) probability for choosing task t_1 and it also determines the complement for choosing task t_2 $(1 - p_1, 1 - p_2)$. The ED predict the changes in behavior of both players.

In Figure 1a, for state 1, the row player receives a reward of 1 independently of the choice of task of the column player. There is hence no incentive for the row player to focus on one of the tasks. The resulting attractor in the ED is a mixed strategy $(0.5, 0.5)$ where the choice of task execution is entirely random as the column player likewise follows suit. If however there is a bias in payoff like in Figure 1b for state 2, there is a shift in preference of the row player. In the second case, task t_2 gives an average higher reward and the row player will

converge to this task and the column player converges to task t_1. Mirrored results hold for the mirrored scenarios of states 3 and 4 from table 1b (not shown) for the column player.

These phenomena are more strongly pronounced if the temperature is increased as can be seen in Figure 1c and d for a temperature τ of 10 instead of 2. The ED hence give an indication of the strength of the attractors for the possible states and show what parameter setting to use to produce strong, desired attractors. In this case, a temperature of $\tau = 2$ is too low as neither the row or column player when seen as a Q-learner is then expected to converge to a strategy where deterministically a unique task per agent is chosen. Both players are expected to converge to a mixed strategy of $\approx (0.83, 0.17)$ (or $(0.17, 0.83)$) and full utility will not be achieved by the MAS on average.
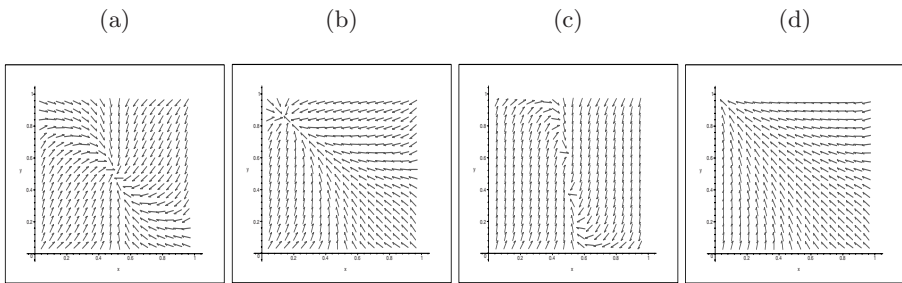
(a)	(b)	(c)	(d)

Fig. 1. The direction field plots of the Dispersion Game with $\tau = 2$ for the row player states 1 and 2 ((a) and (b)), and for $\tau = 10$ ((c) and (d))

Table 1c gives the average expected payoff for the joint actions of the row and column agent. Figure 2 shows the ED of this averaged payoff for $\tau = 10$. This figure is equivalent to the ED of Table 1a for equiprobable choice of states from Table 1b . The row player, and conversely also the column player, initially play an equilibrium strategy where both tasks are equally likely if the system is started with all preferences of all agents for all tasks identical, i.e. point $(0.5, 0.5)$ in figure 2a. A slight bias in preference by one of the agents for either task will however likely push the system away from this weak equilibrium as one of the agents, say the row player, chooses more than average one of the tasks in exploration, say t_1. This means that for the column player, any shift towards a preference for task t_2 will result in a strengthening of this shift. As the row player moves along one of the diagonals away from the mixed equilibrium strategy of $(0.5, 0.5)$, towards 0 or 1 along the x-axis, the column player will have an incentive to move towards 1 or 0 along the y-axis respectively. More precisely, if the row player gets a bias towards task t_1, and thus towards a preference of a 100% chance of choice of task t_1, the column player will develop a bias towards task t_2, and thus towards a preference of 0 for task t_1 (or a preference of a 100% of chance of choosing task t_2) and the dynamics end in point $(1, 0)$ of Figure 2a.

There are now three Nash Equilibriums: $\{(1, 0), (0, 1)\}, \{0, 1), (1, 0)\}$, and $\{(0.5, 0.5), (0.5, 0.5)\}$. However, only the first two are stable as can be seen from the plot in Figure 2a. The mixed equilibrium is very unstable as any small per-

turbation away from this equilibrium will typically lead the dynamics to one of the 2 pure equilibriums. Note that this Dispersion Game is similar to the battle of the sexes game [5, 23].

In Figure 2b we plot a typical Q-learning process using Equation 1 for the above game with the same settings as for the system of differential equations with the sufficiently high temperature of $\tau = 10$ as predicted by the ED. For **only** the row player, the x-axis represents the chance of choosing task t_1 while the y-axis represents the (complementary) chance of choosing task t_2 as defined by the Q-values for these two tasks and the Boltzmann distribution of Equation 4.

As starting points for the Q-learning process we chose a grid of 25 representative points that are chosen so as to keep the plots clear and well-organized. In every point a learning path starts and converges to a particular point where the MAS achieves full utility as the column player develops mirrored preferences. The direction field plots for the value of $\tau = 10$ of Figure 2a predict the expected sample paths of the learning process in the RL domain. The instability of the mixed equilibrium is the reason why this equilibrium does not emerge from the sampled learning process. Of interest for future work is to examine how well the ED can be used to predict the learning trajectories of the Q-learners when an adaptive temperature τ is used in the course of learning as in [4], or how well the ED can predict the impact of learning parameters other than the temperature.

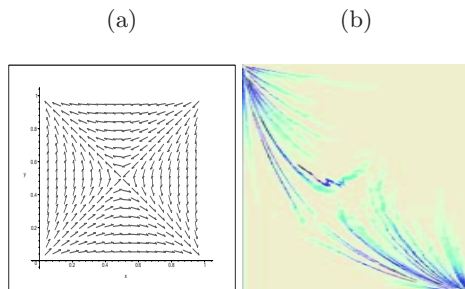(a)   (b)

Fig. 2. The Evolutionary Dynamics and Q-learner with $\tau = 10$

4   COllective INtelligence

In this Section, we briefly outline the theory of COllective INtelligence (COIN) as developed by Wolpert et al., e.g. [26, 24]. Broadly speaking, COIN defines the conditions that an agent's private utility function has to meet to increase the probability that learning to optimize this function leads to increased performance of the collective of agents. Thus, the challenge is to define suitable private utilities function for the individual agents, given the performance of the collective.

Formally, let ζ be the joint moves of all agents. A function $G(\zeta)$ provides the utility of the collective system, the *world utility*, for a given ζ. The goal is to find a ζ that maximizes $G(\zeta)$. Each individual agent η has a private utility function g_η that relates the reward obtained by the collective to the reward that the individual agent collects.

Following a mathematical description of this issue, Wolpert et al. propose the **Wonderful Life Utility** (WLU) as a private utility function that is both *learnable* and *aligned* with G, and that can also be easily calculated.

$$WLU_\eta(\zeta) = G(\zeta) - G(CL_{S_\eta^{eff}}(\zeta)) \tag{8}$$

The function $CL_{S_\eta^{eff}}(\zeta)$ as classically applied "clamps" or suspends the choice of task by agent η and returns the utility of the system without the effect of agent η on the remaining agents $\hat{\eta}$ with which it possibly interacts. For our problem domain, the clamped effect set are those agents $\hat{\eta}$ that are influenced in their utility by the choice of task of agent η.

For example, if agent η picks a task τ which is not chosen by the other agents, the first term of Equation 8 increases appropriately while the second term remains unchanged. Agent η hence receives a reward of $V(\tau)$, where V assigns a value to a task τ. Here, this reward is equal to 1. If the task chosen and executed by η is however also chosen by any of the other agents, the first term $G(\zeta)$ of Equation 8 is unchanged (at least one of the agents executes this task). However, the second term can increase with the value of $V(\tau)$ as agent η "no longer" competes for completion of the task when it is clamped. Agent η then, according to the WLU utility, receives a penalty $-V(\tau)$ for competing for a task targeted by one of the other agents $\hat{\eta}$. The WLU hence has a built in incentive for agents to find an unfulfilled task and hence for each agent to strive for a high global utility in its search for maximizing its own rewards.

The mathematical analysis of the WLU is calculated for the 2 agent full Dispersion Game in payoff Table 3a. Payoffs for the choice of an identical task by the agents in the system are no longer stochastic but are interpreted as a penalty. Behaviors where agents do not interfere in the pursuit of global utility, for example the row agent chooses task t_1 and the column agent chooses task t_2, however receive the original full payoff. Note that the Q-learner update rules of Section 2 are not changed and the agents still act as if they optimize their immediate (discounted) reward. An implementation of a distributed RL system can hence be reused.

5   Evolutionary Dynamics for COIN

The full Dispersion Game has $n!$ stable equilibriums where each of the n agents each fulfills one unique task. These equilibriums coincide with the states of the system that provide full utility. These states are unfortunately increasingly difficult to find in the RL exploration phase as the full joint action state space of the agents is of size n^n. Straightforward use of RL for 100 or more agents (not shown) leads to a maximum performance of ≈ 0.8 as agents prematurely converge to a subset of the tasks [18]. As can be seen in Figure 3b, the COIN framework with the WLU is however able to efficiently solve the assignment problem as the system is scaled whereas classical RL fails. The y-axis shows the utility of the system (1 represents execution of all tasks, utility averaged over 30
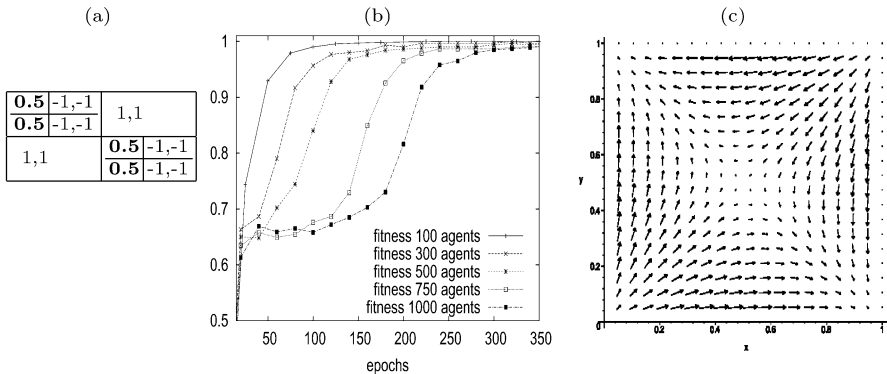
Fig. 3. Details WLU

runs) and the x-axis shows the number of required epochs for the Q-learners of Section 2.

The added value of the COIN framework can be explained by a closer study of the the ED for 2 agents. In Figure 3c we show the differences in the ED with respect to the original ED of the stochastic payoff of Figure 2a for a temperature of $\tau = 10$. This vector field shows where the increased dynamics lead to and the size of the individual vectors show how large the differences are. The differences for just 2 agents are slight, but indicative of the increasing added value as the MAS is scaled. The penalties imposed by the WLU strongly push the MAS away from states where agents choose to execute the same tasks. Aggregations of agents converging on the same task are quickly dispersed in the COIN framework. The ED then indicate that then the agents are on route to desired, optimal and stable Nash equilibriums. Straightforward RL however misses this built in incentive and only improves marginally on the initial random behavior of the system as the number of considered agents is increased. ED plots (not shown) for classical RL where the column agent plays the role of the other agents in the MAS, show that the agents in the MAS on average orbit around the homogeneous preference for all tasks and hence suffer from policy oscillations or converge to a task that is already targeted by another agent.

In further favor of COIN, the added value of the WLU as observed in the ED plots as compared to the standard RL was found to increase as the temperature of the system was increased. This sensitivity analysis, along with the found ED and corresponding attractors give an indication of what settings for the Q-learners will lead to a MAS with a high global utility. As such, the ED are a possible venue to investigate to acquire an indication of viable parameters for COIN and RL, in this case Q-learners.

6   Conclusion

In this work, we presented a novel approach to analyze Reinforcement Learning (RL) for Multi-Agent Systems (MASs). Learning in MASs is a growing area of research. How to achieve a high global utility on a cooperative task is however

still an open question. This work shows a step forward by linking the field of Evolutionary Game Theory (EGT) and engineering approaches for Reinforcement Learning for MASs.

The dynamics of the RL learning process are visualized and analyzed from a perspective of Evolutionary Dynamics. We show how the ED from EGT can be used as a model for Q-learning in stochastic games. We show a one-to-one connection between the behavior of the RL and the ED from evolutionary game theory. Analysis of the evolutionary stable strategies and attractors of the derived ED from the RL application predict the parameters for the RL algorithms to achieve desired Nash equilibriums with high utility.

Secondly, we show how the derived parameter settings from the ED can support application of the COllective INtelligence (COIN) framework. COIN is a proved engineering approach for successful learning of cooperative tasks in MASs. The utilities of the agents are re-engineered to contribute to the global utility. We show that the derived link between ED and RL predicts performance of the COIN framework and visualizes the incentives provided in COIN toward cooperative behavior.

The above approach is potentially viable to apply to other MAS learning methodologies, not just COIN. The one-to-one mapping between the predicted behavior from RD and the observed behavior of the RL'ers is a guide in the design of a MAS.

References

1. B. Banerjee and J. Peng. Adaptive policy gradient in multiagent learning. In *AAMAS*, 2003.
2. A. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete-Event Systems journal, Special issue on Reinforcement Learning*, 13:41–77, 2003.
3. R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman. Transition-independent decentralized Markov decision problems. In *AAMAS*, 2003.
4. C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI/IAAI*, pages 746–752, 1998.
5. C. Gintis. *Game Theory Evolving*. Princeton University Press, 2000.
6. T. Grenager, R. Powers, and Y. Shoham. Dispersion games: general definitions and some specific learning results. In *AAAI 2002*, 2002.
7. C. Guestrin, D. Koller, C. Gearhart, and N. Kanodia. Generalizing plans to new environments in relational MDPs. In *International Joint Conference on Artificial Intelligence (IJCAI-03)*, 2003.
8. J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.
9. P. Huang and K. Sycara. Multi-agent learning in extensive games with complete information. In *AAMAS*, 2003.
10. H. Jung and M. Tambe. Performance model for large scale multiagent systems. In *AAMAS*, 2003.
11. M. Lauer and M. Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proc. 17th International Conf. on Machine Learning*, pages 535–542. Morgan Kaufmann, San Francisco, CA, 2000.

12. M. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, Cambridge, MA, 1994.
13. L. Samuelson. *Evolutionary Games and Equilibrium Selection*. MIT Press, Cambridge, MA, 1997.
14. T. Schneider. Evolution of biological information. journal of NAR, volume 28, pages 2794 - 2799., 2000.
15. D. Stauffer. Life, love and death: Models of biological reproduction and aging. Institute for Theoretical physics, Köln, Euroland, 1999.
16. R. Sutton and A. Barto. *Reinforcement learning: An introduction*. MIT-press, Cambridge, MA, 1998.
17. P. 't Hoen and S. Bohte. COllective INtelligence with sequences of actions. In *14th European Conference on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 2837. Springer, 2003.
18. P. 't Hoen and S. Bohte. COllective INtelligence with task assignment. In *Proceedings of CDOCS03, fortcoming. Also available as TR*, Lecture Notes in Artificial Intelligence. Springer, 2003.
19. K. Tumer and D. Wolpert. COllective INtelligence and Braess' paradox. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 104–109, Austin, Aug. 2000.
20. K. Tuyls, D. Heytens, A. Nowe, and B. Manderick. Extended replicator dynamics as a key to reinforcement learning in multi-agent systems. In *ECML*, Lecture Notes in Artificial Intelligence, LNAI 2837, 2003.
21. K. Tuyls, K. Verbeeck, and T. Lenaerts. A selection-mutation model for Q-learning in multi-agent systems. In *AAMAS*, The ACM International Conference Proceedings Series, 2003.
22. Watkins and Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.
23. J. Weibull. *Evolutionary Game Theory*. The MIT Press, Cambridge, 1995.
24. D. Wolpert and K. Tumer. Optimal payoff functions for members of collectives. *Advances in Complex Systems*, 4(2/3):265–279, 2001.
25. D. H. Wolpert, K. Tumer, and J. Frank. Using collective intelligence to route internet traffic. In *Advances in Neural Information Processing Systems-11*, pages 952–958, Denver, 1998.
26. D. H. Wolpert, K. R. Wheeler, and K. Tumer. General principles of learning-based multi-agent systems. In O. Etzioni, J. P. Müller, and J. M. Bradshaw, editors, *Proceedings of the Third Annual Conference on Autonomous Agents (AGENTS-99)*, pages 77–83, New York, May 1–5 1999. ACM Press.