The Knowledge Engineering Review, Vol. 00:0, 1–1. © 2012, Cambridge University Press
DOI: 10.1017/S000000000000000 Printed in the United Kingdom

A Reinforcement Learning Approach to Coordinate Exploration with Limited Communication in Continuous Action Games

ABDEL RODRÍGUEZ

Computational Modeling Lab, Vrije Universiteit Brussel, Brussels, Belgium
E-mail: abrodrig@vub.ac.be

PETER VRANCX

Computational Modeling Lab, Vrije Universiteit Brussel, Brussels, Belgium
E-mail: pvrancx@vub.ac.be

RICARDO GRAU

Center of Studies in Informatics, Universidad Central "Marta Abreu" de Las Villas, Villa Clara, Cuba
E-mail: rgrau@uclv.edu.cu

ANN NOWÉ

Computational Modeling Lab, Vrije Universiteit Brussel, Brussels, Belgium
E-mail: ann.nowe@vub.ac.be

Abstract

Learning automata are reinforcement learners belonging to the class of policy iterators. They have already been shown to exhibit nice convergence properties in a wide range of discrete action game settings. Recently, a new formulation for a Continuous Action Reinforcement Learning Automata (CARLA) was proposed. In this paper we study the behavior of these CARLA in continuous action games and propose a novel method for coordinated exploration of the joint-action space. Our method allows a team of independent learners, using CARLA, to find the optimal joint action in common interest settings. We first show that independent agents using CARLA will converge to a local optimum of the continuous action game. We then introduce a method for coordinated exploration which allows the team of agents to find the global optimum of the game. We validate our approach in a number of experiments.

1 Introduction

In this paper we present a reinforcement learning technique based on Learning Automata (LA). LA are policy iterators, which have shown good convergence results in discrete action games for independent learners. The approach presented in this paper allows LA to deal with continuous action spaces.

Thathachar and Sastry (2004) introduced a formulation of an LA for continuous action environments known as Continuous Action Learning Automata (CALA). This method uses a parametric probability density function to represent the strategy of the agent. Since the representation is parametric, updating the strategy does not represent an issue from the computational point of view. The CALA implementation is inefficient from a sampling point of view though, since it needs to sample both the selected action and the action corresponding to the current mean of the strategy. This requirement however, is infeasible in many practical settings. An alternative implementation of LA for continuous action games is the Continuous Action Reinforcement Learning Automaton (CARLA), introduced by Howell et al. (1997).

The original CARLA implementation has no sampling drawback, but it is much more demanding from the computational point of view since it does not use a parametric representation. Recently, Rodríguez et al. (2011) reported an improvement of the CARLA method in terms of computation effort and local convergence properties. The improved automaton performs very well in single agent problems, but cannot guarantee convergence to the global optimum in multi-agent settings.

Conventional controller design problems assume that all the controllers present in the system have access to the global information. However, since systems tend to become more and more complex, distributed control entered the picture. A distributed control application assumes there exist several subsystems, each one being controlled by an independent controller preferably sensing only local information. As examples of applications we may refer to formation stabilization, supply chains, wind farms or irrigation channels. While each application has its own specificities, the general idea is that the subsystems with their basic controllers should be brought together in a kind of plug and play manner and still operate successfully. Since these subsystems might have complex interactions this is not trivial. It is also common in conventional control to build a model which is a linearized abstraction of the real plant, calculate the optimal controller and use it in the real plant. Although this approach has successfully been applied in practice, there are plenty of applications where linearizing the model is not possible.

Recently, Vrabie et al. (2009) proposed a model free RL approach for distributed control. The controllers for each subsystem apply an Adaptive Heuristic Critic technique which allows the system to converge to the Nash Equilbrium. It is important however to notice that this approach only works for systems limited to linear interactions. If the system is not linear, the controllers might deviate and never converge. So they are not suitable for non-linearly interacting subsystems.

The CARLA algorithm has successfully been used in control applications (Howell et al., 1997; Howell and Best, 2000). We may expect this algorithm to converge to the unique optimum when the system exhibits linear dynamics and a linear cost function is considered. If the system dynamics or its cost function are not linear, multiple optima may exist. If we use the standard CARLA algorithm for controlling each subsystem in a nonlinear interacting system while ignoring the presence of the other controllers, the system will successfully converge, however convergence to the global optimum cannot be guaranteed. To obtain a better exploration of the action space of the controllers, we propose a technique inspired on the Exploring Selfish Reinforcement Learning approach.

Exploring Selfish Reinforcement Learning (ESRL), introduced by Verbeeck (2004), is an exploration method for a set of independent LA playing a repeated discrete action game. The basic idea of this method is that the set of independent LA will converge to one of the Nash Equilibria (NE) of the game, but not necessarily one from the Pareto front. ESRL proposes that once the agents converge to a Nash equilibrium, at least 2 agents should delete their action which is involved in the selected Nash Equilibrium from their individual action spaces and restart learning. Since the agents are now using a reduced action set, they will converge to a different equilibrium point. Repeating this procedure allows the agents to find all dominant equilibria and agree on the best one, using only limited communication. The learning process is independent, each agent applies the LA learning algorithm, using its individual reward and without knowing which actions have been selected by the other agents. This allows to keep the learning in the individual actions spaces. Only after convergence, a communication protocol takes care of the exclusion of the appropriate actions. As the more interesting Nash equilibria are often also stronger attractors, the agents can quite efficiently reach Pareto optimal Nash equilibria, without the need to exhaustively explore all NE. The problem with applying this approach in continuous action games, is that it makes no sense for the agents to delete the action to which they have converged. Rather, a vicinity around the action should be identified and excluded.

This paper introduces an exploration method for LA in continuous action games inspired by the ESRL method. Section 2 will introduce necessary background information. Standard definitions of random variables and their numerical properties (Parzen, 1960) will be used. The convergence analysis of CARLA when playing continuous action games will be introduced in Section 3 and the need for a better exploration of the joint-action space will be demonstrated. The extension of the discrete action ESRL method to continuous action games will then be introduced in Section 4 showing an adequate way of identifying

the appropriate action range to exclude. This range is identified using only information that is local to the agents. Finally, Sections 5 and 6 will show the experimental results and conclusions.

2 Background

Three important topics will be addressed in this Section. First, we will refer to some standard definitions when dealing with continuous action games in Subsection 2.1, then the CARLA method is introduced in Subsection 2.2 and finally we summarize ESRL in Subsection 2.3.

2.1 Continuous action games

In this Section we introduce some definitions that are necessary before introducing our approach. We will refer to the action space of agent i as $A^{(i)}$ so the joint-action space on n players is $A = A^{(1)} \times \cdots \times A^{(n)}$. In this paper the action spaces are assumed to be compact subsets of $\Re$[1]

The strategy of each player is a probability measure on $A^{(i)}$ and will be denoted as $f^{(i)}\left(a^{(i)}\right)$ for all $a^{(i)} \in A^{(i)}$. We will refer to the space of all possible strategies by $\mathcal{F} = \mathcal{F}^{(1)} \times \cdots \times \mathcal{F}^{(n)}$. The player's feedback or reward is denoted by $\beta^{(i)}\left(a^{(1)}, \cdots, a^{(n)}\right)$. Our work will focus on common interest games, so we can simply refer to the reward as $\beta\left(a^{(1)}, \cdots, a^{(n)}\right)$. Although all agents share the same reward function, the expected reward is not the same for all of them since this expectation is dependent on their individual strategies. The expected reward for a given player with respect to the policies of the other agents $\bar{\beta}^{(i)}\left(a^{(i)}\right)\left|f^{(1)}, \cdots, f^{(i-1)}, f^{(i+1)}, \cdots, f^{(n)}\right.$ can be calculated as shown in (1). Notice that this expectation for every agent i is based on the strategy of all other agents. When we refer to this expected value given the current strategies of the players we will use the short notation $\bar{\beta}^{(i)}\left(a^{(i)}\right)$.

$$\begin{array}{c} \bar{\beta}^{(i)}\left(a^{(i)}\right)\left|f^{(1)}, \cdots, f^{(i-1)}, f^{(i+1)}, \cdots, f^{(n)}\right. = \\ \int_{A^{(1)}} \cdots \int_{A^{(i-1)}} \int_{A^{(i+1)}} \cdots \int_{A^{(n)}} \beta\left(a^{(1)}, \cdots, a^{(n)}\right) df^{(n)}\left(a^{(n)}\right) \\ \cdots, df^{(i+1)}\left(a^{(i+1)}\right) df^{(i-1)}\left(a^{(i-1)}\right) \cdots, df^{(1)}\left(a^{(1)}\right) \end{array} \tag{1}$$

The expected reward of an agent i when using strategy $f^{(i)}$ is defined as (2).

$$\bar{\bar{\beta}}^{(i)}\left|f^{(i)}\right. = \int_{A^{(i)}} \bar{\beta}^{(i)}\left(a^{(i)}\right) df^{(i)}\left(a^{(i)}\right) \tag{2}$$

As the counter part to a Nash equilibrium in discrete action games we use locally superior strategies (Veelen and Spreij, 2009) for continuous action games which is formally introduced in definition 2.1. A vicinity is defined based on a function $d : \mathcal{F}[A] \times \mathcal{F}[A] \to [0, \infty]$ of which we only assume that $d(f, f) = 0$ and $\forall_{f,q \in \mathcal{F}, f \neq q} : d(f, q) > 0$, which implies that d not necessarily is a distance. V is a vicinity of f with respect to d if and only if there is a $\delta > 0$ such that $V = \{q \in \mathcal{F} | d(q, f) < \delta\}$.

Definition 2.1 *f is locally superior with respect to d if it has a vicinity V with respect to d such that* $\forall_{q \in V, q \neq f, i} : \bar{\bar{\beta}}^{(i)}\left|f^{(i)}\right. > \bar{\bar{\beta}}^{(i)}\left|q^{(i)}\right.$

2.2 Continuous action reinforcement learning automata

A learning automaton is a simple model for adaptive decision making in unknown random environments. The original concept of an LA originated in the domain of mathematical psychology (Bush and Mosteller, 1955) where it was used to analyze the behavior of human beings from the viewpoint of psychologists and biologists (Hilgard, 1948; Hilgard and Bower, 1966).

[1] This is a common assumption in control applications.

The engineering research on LA started in the early 1960's (Tsetlin, 1961, 1962). Tsetlin and his colleagues formulated the objective of learning as the optimization of the following performance index[2] (Thathachar and Sastry, 2004; Tsypkin, 1971, 1973):

$$\bar{\beta} = \int_A \beta(a)\, \mathrm{d}f(a) \tag{3}$$

The performance index $\bar{\beta}$ is the expectation of β with respect to the distribution f. This distribution includes randomness in a (probabilistic action selection according to some probability density function (PDF) f) and randomness in β (as rewards can be stochastic). The only assumption on A is that it has to be a compact set. The goal of the learning is to update the probability density function such that the performance measure is optimized.

LA are useful in applications that involve optimization of a reward function which is not known to the learner in the sense that only noise corrupted values of the function for any specific value of actions are observable (Thathachar and Sastry, 2004). A standard implementation is introduced below.

The implementation we are going to use in this paper is the Continuous Action Reinforcement Learning Automata (CARLA) (Howell et al., 1997) because of its applicability to real-world problems (Rodríguez et al., 2011). In Howell et al. (1997) f is implemented as a nonparametric probability density function (PDF). Starting with the uniform distribution over the whole action space A and after exploring action $a_t \in A$ in time step t the PDF is updated as shown in (4). Parameters α and λ are referred to as learning and spreading rate respectively. The first one, α, determines the importance of new observations, the higher the rate the larger the change in the PDF from one time-step to another. The second parameter, λ, determines how much to spread the information of a given observation to the neighboring actions. Finally, γ is a normalization factor so $\int_{-\infty}^{+\infty} f_{t+1}(z)\, dz = 1$

$$f_{t+1}(a) = \begin{cases} \gamma_t \left(f_t(a) + \beta_t(a_t)\, \alpha e^{-\frac{1}{2}\left(\frac{a-a_t}{\lambda}\right)^2} \right) & a \in A \\ 0 & a \notin A \end{cases} \tag{4}$$

Figure 1 shows a sketch of the CARLA action space exploration. At time-step t_0 (Figure 1.a) the learner's strategy f is the uniform PDF and action a_0 is drawn at random. Then f is updated by adding a Gaussian bell around a_0. The height is determined by the reward β_t and the width by the spreading parameter λ. Then the PDF is normalized again as shown by the bold line in Figure 1.b. From the new strategy f the action a_1 is drawn and the strategy is again updated as shown in 1.c, the resulting PDF is shown in bold. In this example both actions (a_0 and a_1 received positive rewards so the probabilities of the neighboring actions were positively reinforced).

Every time the agent selects an action, this action will be drawn stochastically according to the PDF. Therefore the cumulative density function (CDF) is required (Parzen, 1960). Since the PDF used in the CARLA method is nonparametric, it is computationally expensive to generate the CDF every time the function changes. The method used in this paper for the CDF is the approximation introduced by Rodríguez et al. (2011) shown in expression (5). Notice that $F_{N(0,1)}$ is the standardized normal cumulative density function. Also notice that the λ parameter is now dynamically changing over time. This is also introduced by Rodríguez et al. (2011) as an improvement of the method to speed-up the convergence. The spreading rate used at every time-step is calculated as a factor of the original parameter λ_0 multiplied by a convergence measure based on the standard deviation of the last n actions σ_t. By directly using the CDF, numerical integration is avoided so the method becomes feasible in terms of computation. By dynamically updating the spreading rate, the convergence to the local optima is accelerated and the method is improved in terms of exploration.

$$F_{t+1}(a) = \begin{cases} 0 & a < \min(A) \\ \gamma_t \left(F_t(a) + \delta_t D_t(\min(A), a) \right) & a \in A \\ 1 & a > \max(A) \end{cases} \tag{5}$$

[2]The original formulation is $J(\Lambda) = \int_\Lambda R(\mathbf{a}, \Lambda)\, \mathrm{d}f(\mathbf{a})$ but for consistency reasons, we adapted this formulation to the notation used in this paper.

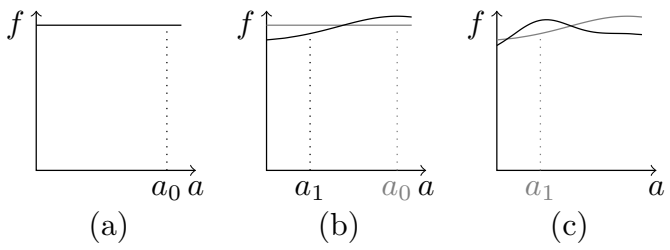

Figure 1: Sketch of CARLA action space exploration. The solid line represents the current PDF while the gray line represents the previous PDF

where:

$$\delta_t = \beta_t(a_t)\,\alpha\lambda_t\sqrt{2\pi}$$

$$\gamma_t = \frac{1}{1+\delta_t D_t(\min(A),\max(A))}$$

$$D_t(x,y) = F_{N(0,1)}\left(\frac{y-a_t}{\lambda_t}\right) - F_{N(0,1)}\left(\frac{x-a_t}{\lambda_t}\right)$$

$$\lambda_t = \lambda_0 \frac{12\sigma_t}{(\max(A)-\min(A))^2}$$

2.3 Exploring selfish reinforcement learners

In this Subsection we will briefly explain exploring selfish reinforcement learning which was introduced for efficient exploration of the NE for reinforcement learners with discrete action spaces[3] (Verbeeck, 2004). The main characteristic of ESRL is the way by which the agents coordinate their exploration. Phases in which agents act independently and behave as selfish optimizers (exploration phase) are alternated by phases in which agents are social and act so as to optimize the group objective (synchronization phase). During every exploration phase the agents individually apply an LA learning scheme and will converge to a pure Nash equilibrium. In the synchronization phase the solution found at the end of the exploration phase is evaluated and removed from the joint-action space. In this way, new attractors can be explored. The removal is achieved by letting at least two agents exclude their private action that is part of the joint solution. Agents alternate between exploration and synchronization phases to efficiently search the shrinking joint-action space in order to find the Pareto optimal Nash equilibrium.

In common interest games at least one Pareto optimal Nash equilibrium exists. Despite the existence of these optimal solutions, there is no guarantee that independent learners converge to such an equilibrium. Only convergence to (possibly suboptimal) Nash equilibria can be guaranteed (Claus and Boutilier, 1998). Games such as the climbing game (Kapetanakis et al., 2003), shown in Figure 2 are generally accepted as hard coordination problems for independent learners[4]. In this game, independent learners may get stuck in the suboptimal Nash equilibrium (a_{12},a_{22}) with payoff 7. Whether the agents reach the Pareto optimal Nash equilibrium with payoff 11 mainly depends on the initialization of the action probabilities when learning starts. Independent learning agents will have difficulties reaching the optimal solution because of the heavy penalties surrounding it.

[3] A a conflicting interest version of ESRL also exists, however, since we only use common interest settings in this paper, we only describe the common interest version

[4] ESRL can perfectly deal with games with stochastic payoff but we illustrate the idea on a deterministic example

	a_{21}	a_{22}	a_{23}
a_{11}	11	-30	0
a_{12}	-30	7	6
a_{13}	0	0	5

Figure 2: The climbing game, a common interest game. The Pareto optimal Nash equilibrium (a_{11}, a_{21}) is surrounded by heavy penalties

	a_{21}	a_{23}
a_{11}	11	0
a_{13}	0	5

Figure 3: The climbing game after shrinking the joint-action space. The Pareto optimal Nash equilibrium (a_{11}, a_{21}) is still not the only one Nash equilibrium but it is now the most likely to find

If the Pareto optimal Nash equilibrium is reached, there is no room for improvement. However if this Pareto optimal Nash equilibrium is not reached, all agents will benefit from searching for it in the long run. Suppose for example that the agents of Figure 2 have converged to joint action (a_{12},a_{22}) corresponding to a payoff of 7 for both agents. Independent learners will not have the tendency to explore further to reach the better Nash equilibrium, with payoff 11 for both agents, because of the high punishment of -30 that surrounds this interesting Nash equilibrium.

In order not to accumulate too much punishment before discovering the best joint action (a_{11},a_{21}), the agents should coordinate their exploration. This means that the agents should decide together to search for a better Nash equilibrium. In ESRL this is achieved by making the agents exclude the action they converged to and then restarting learning in the remaining action subspace. In the case of the climbing game presented above, the joint-action space shrinks from 9 to 4 joint actions as shown in Figure 3. Again the learning itself can be performed in an independent manner, and this will drive the agents toward another Nash equilibrium. The average payoff received during this second period of learning can be compared with the average payoff received during the first period of learning, and as such the two Nash equilibria found can be compared against each other. This alternation between independent learning and excluding actions can be repeated until one agent has only one action left. This approach has been validated in much bigger games (Verbeeck, 2004) with more agents and more actions, as well as stochastic payoffs. Moreover the approach has also shown to beneficial in cases where agents take actions asynchronously and delayed rewards, such as for example in job scheduling settings, where agents take actions when a job is generated and jobs first enter a queue before being processed.

3 Convergence of CARLA

The convergence proof for a single agent CARLA was introduced by Rodríguez et al. (2011). A similar analysis for continuous action games is given in Appendix A. We show that a set of CARLA will converge to a local attractor in the joint-action space as long as the parameter α is sufficiently low for the learners to match their expected strategies through time. Which of the multiple attractors is selected will depend on the initialization of the parameters.

In order to better understand, let us analyze the following example. The analytical expression of the reward signal is shown in (21) in Appendix B under the name β and its contour representation in Figure 4. There are three attractors in this example. The two local maxima located in the top left and bottom right corners have larger basins of attraction while the global maximum at the center has a narrower basin of attraction.

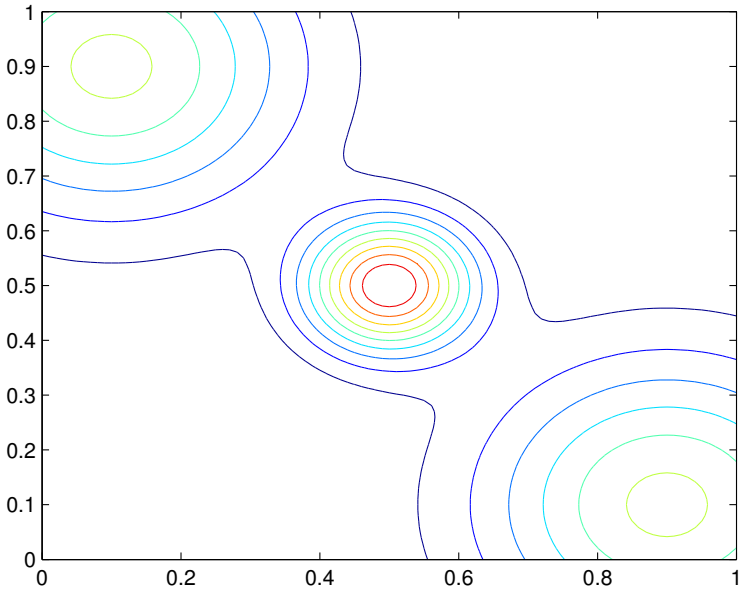

Figure 4: A two player game with three local optima. The contours represents combination of actions with the same reward (also see Appendix B). Notice that the global optimum is located in a narrower region, this region does not overlap with the other maxima from both agents point of view.

As shown in Appendix A if both learners start with the uniform distribution we expect them to converge to the attractor that fits expression (6) the best.

$$\forall_{a\in A^{(i)}} : \int_{-\infty}^{+\infty} \bar{\beta}_t^{(i)}(z)\, e^{-\frac{1}{2}\left(\frac{a_*^{(i)}-z}{\lambda_t^{(i)}}\right)^2} \mathrm{d}z \geq \int_{-\infty}^{+\infty} \bar{\beta}_t^{(i)}(z)\, e^{-\frac{1}{2}\left(\frac{a-z}{\lambda_t^{(i)}}\right)^2} \mathrm{d}z \tag{6}$$

Since we are only interested in demonstrating the effect of λ we use a fixed value over all time-steps. Two λ values will be explored: 0.04 and 0.007. The learners are expected to converge to either one of the attractors to the corners (either (0.1,0.9) or (0.9,0.1)) when using $\lambda = 0.04$ and to the global maximum (0.5,0.5) when $\lambda = 0.007$. In order to prevent the learners from deviating too much from their expected strategies, a very low α value of 0.005 is used in this case. In a real setup we would never use such a low value. Because of the low learning rate we run the 100 experiments for 30000 time-steps. When using $\lambda = 0.04$, half of the times the learners converge to the attractor (0.1,0.9) while in the other half of the cases they converge to (0.9,0.1). In the second setting, when using $\lambda = 0.007$, both agents converge to the global maximum. We would like to stress that selecting a very small spreading rate is not a practical solution. The probability mass added to the explored action not only depends on the reward or the learning rate but also on the spreading rate. The larger the spreading rate, the larger the reinforcement around the selected action is. Selecting very low values of spreading rate will significantly lengthen the learning process. Moreover, the spreading rate controls to what extend the knowledge at a given action is extrapolated over the neighboring actions. Using lower the spreading rates, less information will be extended from one action into their neighbors. Using low values of spreading rate will thus cause the learners to explore the action space in more detail. As a result, using a value that is too low, will also lengthen the exploration unnecessarily.

We can conclude that in order to make the learners converge to the best attractor without extending the learning time too much, we either need information about the reward function or a better exploration of the joint-action space, as we will introduce in the next Section.

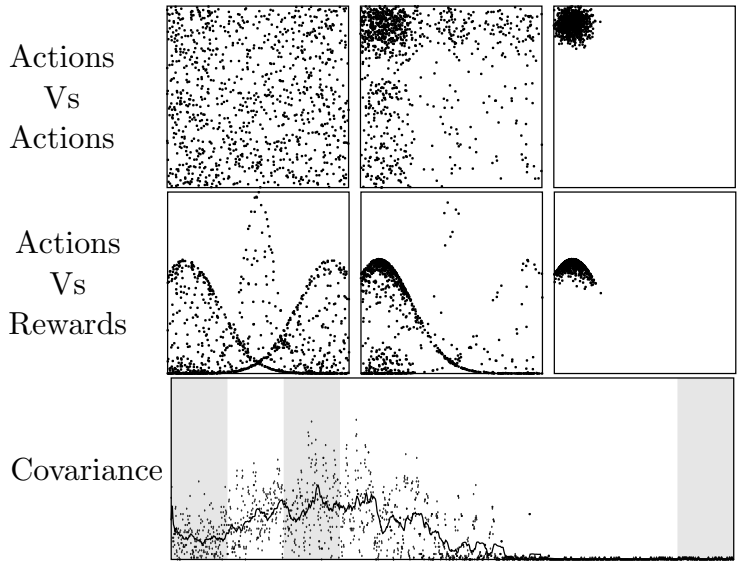

Figure 5: Relation between covariance and exploration. The row on top represents the joint actions selected by the agents. The second row represents the reward obtained by the first agent for each of the selected actions . The left-most column represents the samples from time-step 0 to 1000. The column in the center depicts the samples from time-step 2000 to 3000. The right-most column shows the samples from time-step 9000 to 10000. The last row shows the evolution of the absolute value of the covariance between actions and rewards for the first agent over the 10000 time-steps. The intervals corresponding to 0 to 1000, 2000 to 3000 and 9000 to 10000 are shaded.

4 ESCARLA

In order to introduce Exploring Selfish Continuous Action Reinforcement Learning Automaton (ESCARLA) let us first recall the basic idea of ESRL which is to exclude actions after every exploration phase. The problem when facing continuous action games is that it makes no sense that the agents delete a single action. Instead we consider removing a neighborhood of the action corresponding to the basin of attraction of the joint action to which the learners converged.

In order to identify the basin of attraction, we propose to use the absolute value of the covariance between the actions and rewards as a metric. Figure 5 shows the relation between the exploration and the covariance between actions and rewards from a single agent point of view. The parameter setting used for this example is the same as the one selected in Section 3. The first row shows a global view of the exploration. Three time intervals are shown. The first interval is the beginning of learning (time-steps from 0 to 1000). The second interval is when the learners are leaving global exploration (time-steps from 2000 to 3000). The last interval selected is when agents have converged to the local attractor (time-steps from 9000 to 10000). From the middle top figure, one can notice that the learners are converging to a locally superior strategy and are trapped within its basin of attraction. The dark left top cluster shows that the actions near to the locally superior strategy (0.1,0.9) are now sampled more frequently. The challenge now is to identify this using only information local to the agents. The second row shows the local information the agents have to their disposal. The same time intervals are represented in each respective column. The plots depict the selected actions on the horizontal axis with the corresponding reward on the vertical axis. The bottom row shows the absolute value of the covariance between actions and rewards over the whole learning process. Additionally, in order to have a better idea of how this covariance is evolving, the solid curve represents its average. The three intervals considered above are shaded in gray. This covariance has a low value at the beginning of learning since a lot of exploration is performed by both agents. When the agents start to be influenced by the basin of attraction of a local attractor, the noise

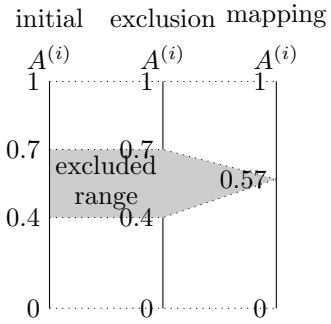

Figure 6: Mapping process. The new action space will stay as the previous one but actions will be mapped in order to avoid the deleted range (0.4,0.7)

in the rewards significantly drops, so the covariance reaches its maximum. As agents converge to a locally superior strategy, less exploration is performed and therefore the covariance value drops down to zero. The appropriate region to exclude after the agent's actions have converged to the local optimum, can therefore be estimated at the moment when the absolute value of the covariance reaches its maximum value.

A good way of estimating this region is by using the percentiles of the probability density function of the actions. For a given confidence value c we can define a region as shown in expression (7) where $\text{percentile}(p,f)$ represents the value where the probability density function f accumulates the probability p. Notice that the lower the c the wider the region defined by expression (7), but also the lower the probability for the actions in the region to have been properly explored.

$$\left[\text{percentile}\left(\frac{1-c}{2},f\right),\text{percentile}\left(1-\frac{1-c}{2},f\right)\right] \tag{7}$$

The proposal is to let the agents learn until they all reach a threshold value of convergence $conv_{stp}$. Notice that any exploration phase will only be over when all agents have converged. Some limited communication is needed for this agreement on the ending of every exploration phase. Then each agent should delete the region defined by (7) from its action space – examples in the experimental results Section will use $conv_{stp} = 0.95$ and $c = 0.9$. Deleting the action range implies modifying the action space so we need to map the remaining action space onto a compact set again as shown in Figure 6. In the example given, the region to exclude is $(0.4, 0.7)$. The new action space will be composed by the two remaining intervals $[0, 0.4]$ and $[0.7, 1]$. To the effects of exploration, this new action space will be linearly mapped into $[0, 1]$. Both of them have a subset on it proportional to their extension for remaining actions should be equally eligible as they were before the mapping.

This method can introduce some abrupt jumps into the reward function as observed by the agents due to the eliminated regions, but since CARLA learners do not require continuity in the reward function this does not cause a problem. After the exclusion of the appropriate action range, all agents must restart the learning process in order to converge to another attractor. After sufficient exploration the agents should compare the reward obtained for the different joint actions to which they converged and agree on the strategy that yielded the highest reward. It is important to remember that we are assuming a common interest game, so therefore the agents have no problem to agree on the best combination of actions.

The general procedure is given in algorithm 1. The exploration and synchronization phases are also given in algorithms 2 and 3 respectively.

5 Experimental results

In this Section we report some experimental results. First some abstract complex continuous action games are considered and then we also evaluate our approach on a more realistic control case.

Algorithm 1 Exploring Selfish CARLA

repeat
  explore
  synchronize
until enough exploration
select best strategy

Algorithm 2 Exploration phase (ESCARLA)

initialize parameters
repeat
  sample action
  update strategy
  if maximum covariance **then**
    interval $\leftarrow \left[\text{percentile}\left(\frac{1-c}{2}, f\right), \text{percentile}\left(1-\frac{1-c}{2}, f\right)\right]$
  end if
until convergence

Algorithm 3 Synchronization phase (ESCARLA)

exclude marked interval

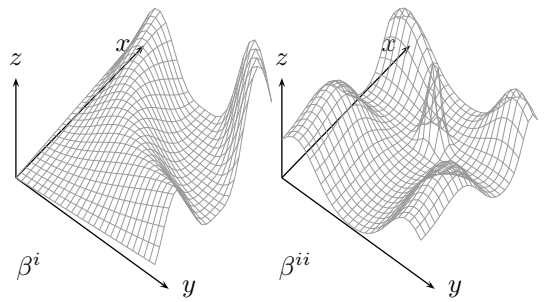

Figure 7: Reward functions of both games. Notice in the first game to the left that the global optimum is located in a very narrow region, but this region overlaps with the other maxima in both axes (first and second agent's axis). The second game to the right is even more complex since it also has a very narrow optimum region but it does not overlap with the other maxima.

It is common to represent a continuous action game with their reward function. Two 2-agent games will be considered next. The common payoff functions are plotted in Figure 7 and the analytical expressions are given in (21) of Appendix B. These two games are introduced as complex examples where the basins of attractions are overlapping from both agent point of views.

Figure 8 shows the learning performance of the agents. More specifically the average rewards collected over time are given in the figure on top and the actions selected every time-step are shown in the picture at the bottom. The parameters were set as follows: α was 0.1 while λ_0 was set to 0.2. This λ_0 value has empirically shown in experiments to give good results. A larger value may cause the learner not to converge while a shorter value lengthen the learning process unnecessarily. The value of α was selected low to avoid unstable behavior of the method. The higher the value the faster the convergence but at the same time, the more unstable the learning. A large value for α might produce early convergence to local optima. In a real world application this parameter has to be chosen more carefully since convergence time is also important. For these experiments we decide to choose a more reliable value at the cost of

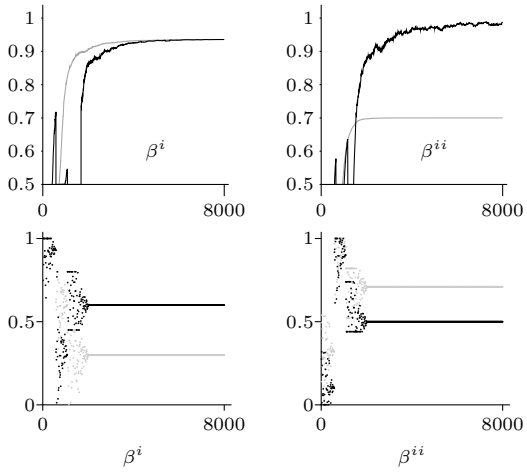

Figure 8: Average rewards for game 1 (to the right)and game 2 (to the left). For the first game, the CARLA method is able to converge to the global optimum, the exploration by ESCARLA is not needed . However, for the second game, the ESCARLA clearly brings an improvement compared to the CARLA method. While the CARLA approach does not bring the agents to the global optimum, the ESCARLA leads the agents to the global optimum in the third exploration phase. Average reward is shown on top while the selected actions are shown on bottom

1^{st} phase	2^{nd} phase	3^{rd} phase	4^{th} phase	total percent
0	0	32	2	68%

Table 1 Convergence to the global optimum per exploration phase

learning time. The gray curve shows the result using the CARLA method. The black curve shows the results when using the ESCARLA. Notice that every peak in the rewards for ESCARLA implies a restart in the learning. Although a bit slower, the ESCARLA obtains more accurate convergence to the global optimum and does not get stuck in a suboptimal solution as CARLA on this second game.

Figure 9 shows the region elimination process for the second game. In phase one the agents converge to the joint action $(0.32, 0.10)$ and delete the region ($[0.04, 0.52]$, $[0.01, 0.44]$) which is shown in black. In the second phase they converge to $(0.95, 0.9)$ and delete the region ($[0.82, 0.99]$,$[0.74, 0.99]$). Finally – phase three –, they easily converge to $(0.71, 0.5)$.

We would like to stress that the basins of attraction of the local attractors are not symmetrical, in other words the roles of agents are not interchangeable. This can be seen by inspecting equation (21) in Appendix B, the function is not symmetrical due to taking the square of the action of the first agent. As a consequence the range of actions excluded by the agents does not have the same size.

Given the stochastic nature of the optimization process and the interaction of both agents we cannot be sure that the agents will always find the global optimum after as many exploration phases as maxima in the game. Two problems may arise, either the global optimum could be excluded with a suboptimal strategy or the range of actions excluded in a synchronization phase could cover just a partial subset of the basin of attraction so probably an extra exploration phase will be necessary. In order to illustrate this issue Table 1 shows how many times the global optimum was found in every exploration phase for the second game after 50 trials. Notice that the CARLA method results are the same as the ESCARLA in the first exploration phase.

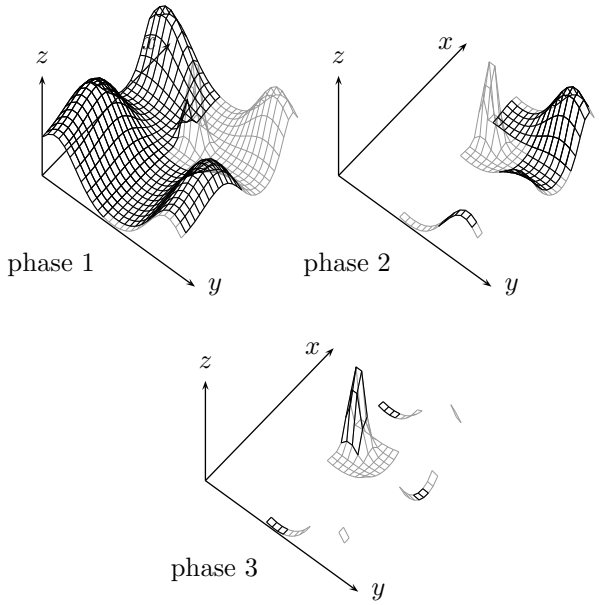

Figure 9: Region elimination process. Notice that it is very unlikely to find the global optimum because it is a very narrow region. After ending the first exploration phase and restarting the learning, it becomes much easier to identify. After the second restart, it is inevitable to find it.

5.1 Statistical significance

The learners perform well in the above games but, to know whether this conclusion can be generalized we evaluate the new approach on randomly generated games.

Random functions are generated for the reward function of the games. The functions are generated by the union of Gaussian bells. In order to demonstrate that the performance of the technique is not lowered some single agent functions are also taken into account. Single agent functions use 1 or 2 bells while multi-agent functions will use between 1 and 6 of these bells. The number of bells is generated at random. The parameters of the bells $\mu \in [0,1]$ and $\sigma \in [0.01,1]$ are generated randomly too, as well as the factor $f \in [0.5,1]$ multiplying the factor for the bells to make them different in amplitude. An extra parameter $r \in [0.75,0.95]$ is generated for introducing noise in single agent settings. Two ways for introducing noise in single agent setting are explored with these functions, adding noisy bells or randomly selecting between them. The analytical expressions are introduced in (21) in Appendix B. Notice that β_1, β_2 and β_3 are single agent problems while β_4 is the only one representing a game of two players.

50 games were generated from each type of function. The average linear difference between the global maximum of the function and the final average reward collected by agents over 10000 time-steps are shown in Table 2. The same parameter settings from the previous tests are used. A Wilcoxon test shows that the ESCARLA performs better than the original method for multi-agent games. The results of this test can be seen in the last column of the Table 2.

5.2 Water reservoirs problem

Below, we report a control problem which is a simplified yet representative problem of controlling the water level of a network of connected reservoirs or locks. Typically such a network is spread over different

	CARLA	ESCARLA	sig.
β_1	0.0049	0.0146	0.352
β_2	0.0092	0.0100	0.746
β_3	0.0178	0.0171	0.515
β_4	0.0851	0.0365	0.001

Table 2 Mean absolute error. Every row shows the average difference between the long-run reward and the current maximum reward. The first row shows the previous CARLA, the second shows results for the ESCARLA method. ESCARLA considerably reduces the difference for multi-agent games while it does not harm performance in single-agent settings. The last column shows the Wilcoxon test significance

countries or authorities who want to decide independently on the level of the reservoirs belonging to their territory, yet it is not allowed to get the other ones in situations that would lead to flooding or water scarcity. An example is the Belgian lock systems. Water enters from France and Germany, and within Belgium the locks located in the South are controlled by Wallonia and the locks in the North by Flanders. Controlling reservoirs by RL is not entirely new. In Castelletti et al. (2012) an RL approach is described for controlling a single reservoir. In this paper the setting is much more complicated as a network of locks is considered.

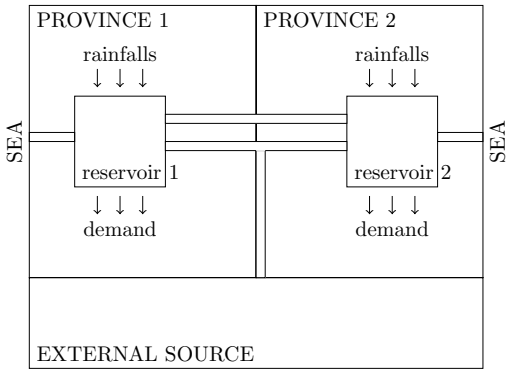

Figure 10: Water reservoirs problem

Consider two water reservoirs that belong to different provinces. Each reservoir covers the demand of the region. The reservoirs are normally filled by rainfall. Both reservoirs may discharge to the sea using a flow limited by means of a valve. The reservoirs are connected so they can transport water from one to the other. This flow comes at a price that linearly depends on the levels of water of each reservoir, flowing from a high level to a lower one is much cheaper than the other way around. In case of emergency they may also use an external source for feeding or discharging the reservoir. This external resource can only be used when both reservoirs are full or empty to avoid flooding or water scarcity. There is only one connection to the external source that becomes more expensive when used by both reservoirs at the same time. Figure 10 illustrates this problem.

A more detailed description of the system is given next. Rain fall is normally distributed with mean 0.5 and standard deviation 0.1 in both provinces. The demand is fixed in both provinces at 0.5. Valves controlling water to flow to the sea of both reservoirs allow a maximum flow of 0.35 per time unit. The cost of transporting a unit of water from or to the external source is 0.5 if it is used by only one province and 4 when used by the both of them. The cost for transporting water from one province to the other depends on the difference of water levels as shown in equation (8) where $level_1$ is the level of the water in the reservoir of the province which is giving the water and $level_2$ is the level of the water of the reservoir of the province which is asking for the water. The objective of this problem is to find the adequate levels of

	1^{st} phase	2^{nd} phase	total percent
$(0,1)$	26	24	50
$(1,0)$	24	26	50
	50	50	100

Table 3 Convergence to each optimum per exploration phase

water that the reservoirs should keep to meet the demand at the lowest cost. If a reservoir's level is higher than the target level, it discharges the water to the other reservoir if the other needs it to reach its target level or to the sea as fast as the valve allows it. The cost is calculated after 50 time units starting from random levels of water at each reservoir. Figure 11 shows a contour representation of the average reward (calculated over a hundred simulations) that the learners may expect when facing this problem (a contour representation to the left and the landscape representation to the right). The expected cost is plotted over the z-axis and the actions runs from completely empty (0) to completely full (1).

$$cost = 50\,(level_2 - level_1 + 1) \tag{8}$$

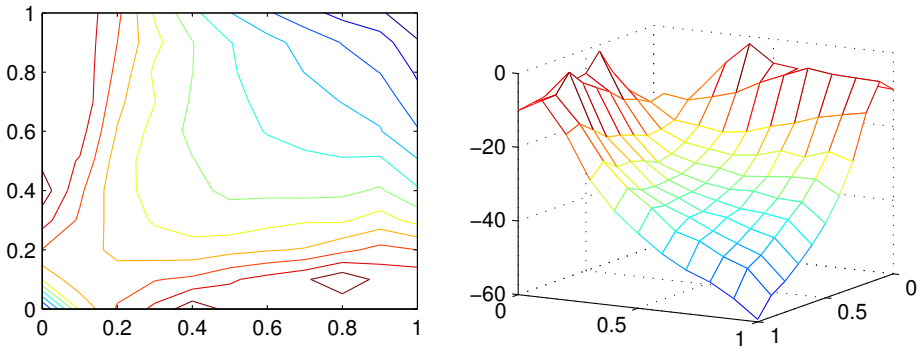

Figure 11: The water reservoirs problem. On the left the contour representation of the reward, while on the right its landscape is shown.

Note that there are two Nash equilibria in this problem. The total amount of water in both provinces has to be sufficient to avoid the need for feeding the reservoirs from the external source but not too much to avoid discharging to the external source at a high cost. This is necessary due to the stochasticity of the rainfall. Because it is much more expensive to simultaneously use the external resource for both reservoirs and it is much cheaper to move water from one reservoir to the other if there is a big difference in their levels it is preferable to store most of water in one reservoir and keep the other lower.

Table 3 shows the percentage of the times that each optimum was found with ESCARLA method at each phase using the same settings as before. Each row represents the percentage of times that the top-left or bottom-right maximum was found. The last row shows that ESCARLA was always successful in finding one of the Nash equilibria in the first phase and the other one in the second phase. Notice that they find both Nash equilibria with equal probability, which is to be expected as the problem is symmetric.

6 Conclusions

In this paper we propose a distributed RL approach which cannot only deal with continuous actions but also with complex interactions. We show that two agents applying independently an LA update scheme can converge to a Nash equilibrium or even the Pareto optimal Nash equilibrium by extending the ESRL idea. More precisely, we introduce a novel method for estimating the basin of attraction around a locally superior strategy in a common interest game. This allows learners to efficiently explore the

action space, by avoiding the basin of attraction of previously discovered attractors. This affirmation is supported by a number of experiments. The new method achieves good results in artificially created functions that represent challenging distributed optimization problems. A control problem inspired by a real world problem is also introduced and it is shown that our approach successfully solves the problem. We believe that this approach has great potential for the control of complex systems, which are hard to solve analytically. Examples are complex networks such as telecom networks or water reservoir networks. In future work we plan to study the effect of this technique in games of more than two agents and in conflicting interest games.

References

R. Bush and F. Mosteller. *Stochastic Models for Learning*. Wiley, 1955.

A. Castelletti, F. Pianosi, and M. Restelli. Tree-based Fitted Q-iteration for Multi-Objective Markov Decision problems. In *IJCNN*, pages 1–8. IEEE, 2012.

C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of National Conference on Artificial Intelligence (AAAI-98*, pages 746–752, 1998.

E. Hilgard. *Theories of Learning*. New York: Appleton-Century-Crofts, 1948.

E. Hilgard and B. Bower. *Theories of Learning*. New Jersey: Prentice Hall, 1966.

M. Howell and M. Best. On-line PID tuning for engine idle-speed control using continuous action reinforcement learning automata. *Control Engineering Practice*, 8(2):147 – 154, 2000. doi: 10.1016/S0967-0661(99)00141-0.

M. Howell, G. Frost, T. Gordon, and Q. Wu. Continuous action reinforcement learning applied to vehicle suspension control. *Mechatronics*, 7(3):263 – 276, 1997. doi: DOI: 10.1016/S0957-4158(97)00003-2.

S. Kapetanakis, D. Kudenko, and M. Strens. Learning to coordinate using commitment sequences in cooperative multiagent-systems. In *Proceedings of the Third Symposium on Adaptive Agents and Multi-agent Systems (AAMAS-03*, page 2004, 2003.

E. Parzen. *Modern Probability Theory And Its Applications*. Wiley-interscience, wiley classics edition edition, December 1960.

A. Rodríguez, R. Grau, and A. Nowé. CONTINUOUS ACTION REINFORCEMENT LEARNING AUTOMATA. Performance and convergence. In J. Filipe and A. Fred, editors, *In Proceedings of the Third International Conference on Agents and Artificial Intelligence*, pages 473–478. SciTePress, January 2011.

M. Thathachar and P. Sastry. *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Kluwer Academic Publishers, 2004.

M. Tsetlin. The behavior of finite automata in random media. *Avtomatika i Telemekhanika*, pages 1345–1354, 1961.

M. Tsetlin. The behavior of finite automata in random media. *Avtomatika i Telemekhanika*, pages 1210–1219, 1962.

Y. Tsypkin. *Adaptation and learning in automatic systems*. New York: Academic Press, 1971.

Y. Tsypkin. *Foundations of the theory of learning systems*. New York: Academic Press, 1973.

M. Veelen and P. Spreij. Evolution in games with a continuous action space. *Economic Theory*, 39(3): 355–376, June 2009.

K. Verbeeck. *Coordinated Exploration in Multi-Agent Reiforcement Learning*. PhD thesis, Vrije Universiteit Brussel, Faculteit Wetenschappen, DINF, Computational Modeling Lab, September 2004.

D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. Lewis. Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica*, 2(45):477–484, 2009.

Appendix A CARLA convergence in continuous action games

The analysis will be performed for normalized reward signals $\beta : \Re \to [0,1]$ – no generality is lost because any closed interval can mapped to this interval by a linear transformation. The final goal of this analysis is to find the necessary restrictions to guarantee convergence to local optima.

The sequence of PDF updates is a Markovian process, where for each time-step t an action $a_t^{(i)} \in A^{(i)}$ is selected and a new $f_t^{(i)}$ is returned for every learner i. The expected value $\bar{f}_{t+1}^{(i)}$ of $f_{t+1}^{(i)}$ can be computed following equation (9).

$$\bar{f}_{t+1}^{(i)}(a) = \int_{-\infty}^{+\infty} f_t^{(i)}(z)\, f_{t+1}^{(i)}\left(a \mid a_t^{(i)} = z\right) dz \tag{9}$$

Let $\bar{\gamma}_t^{(i)}(z) = \bar{\gamma}_t^{(i)} | a_t = z$ be the expected value for $\gamma_t^{(i)}$ if $a_t^{(i)} = z$ taking into account the other agent policies, $\bar{\bar{\gamma}}_t^{(i)}$ its expected value and $\bar{\beta}_t^{(i)}(z)$ the expected value for $\beta_t^{(i)}(z)$ as shown in (10). Then (9) could be rewritten as (11).

$$\begin{aligned}
\bar{\gamma}_t^{(i)}(z) = & \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} f_t^{(0)}\left(z^{(0)}\right) \cdots f_t^{(i-1)}\left(z^{(i-1)}\right) \\
& f_t^{(i+1)}\left(z^{(i+1)}\right) \cdots f_t^{(n)}\left(z^{(n)}\right) \\
& \left(\gamma_t^{(i)} | a_t^{(i)} = z, a_t^{(0)} = z^{(0)}, \cdots a_t^{(n)} = z^{(n)}\right) dz^{(n)} \cdots dz^{(0)} \\
\bar{\beta}_t^{(i)}(z) = & \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} f_t^{(0)}\left(z^{(0)}\right) \cdots f_t^{(i-1)}\left(z^{(i-1)}\right) \\
& f_t^{(i+1)}\left(z^{(i+1)}\right) \cdots f_t^{(n)}\left(z^{(n)}\right) \\
& \beta_t^{(i)}\left(z^{(0)}, \cdots, z^{(i-1)}, z, z^{(i+1)}, \cdots, z^{(n)}\right) dz^{(n)} \cdots dz^{(0)} \\
\bar{\bar{\gamma}}_t^{(i)} = & \int_{-\infty}^{+\infty} f_t^{(i)}(z)\, \bar{\gamma}_t^{(i)}(z)\, dz
\end{aligned} \tag{10}$$

$$\begin{aligned}
\bar{f}_{t+1}^{(i)}(a) = & \int_{-\infty}^{+\infty} f_t^{(i)}(z)\, \bar{\gamma}_t^{(i)}(z) \\
& \left(f_t^{(i)}(a) + \alpha^{(i)} \bar{\beta}_t^{(i)}(z)\, e^{-\frac{1}{2}\left(\frac{a-z}{\lambda_t^{(i)}}\right)^2} \right) dz \\
= & f_t^{(i)}(a)\, \bar{\bar{\gamma}}_t^{(i)} + \\
& \alpha^{(i)} \int_{-\infty}^{+\infty} f_t^{(i)}(z)\, \bar{\gamma}_t^{(i)}(z)\, \bar{\beta}_t^{(i)}(z)\, e^{-\frac{1}{2}\left(\frac{a-z}{\lambda_t^{(i)}}\right)^2} dz
\end{aligned} \tag{11}$$

Let us have a look at the right member of the integral. $f_t^{(i)}(z)$ is multiplied by the factor composed by the normalization factor given that $a_t^{(i)} = z$, the feedback signal $\bar{\beta}_t^{(i)}(z)$ and the distance measure $e^{-\frac{1}{2}\left(\frac{a-z}{\lambda_t^{(i)}}\right)^2}$ which can be interpreted as the strength of the relation of actions a and z, the higher the value of this product, the bigger the relation of these actions. Let us call this composed factor $G_t^{(i)}(a,z)$ and $\bar{G}_t^{(i)}(a)$ its expected value at time-step t with respect to z. Then equation (11) could be finally formulated as (12).

$$\begin{aligned} \bar{f}_{t+1}^{(i)}(a) &= f_t^{(i)}(a)\bar{\bar{\gamma}}_t^{(i)} + \alpha^{(i)}\bar{G}_t^{(i)}(a) \\ &= f_t^{(i)}(a)\left(\bar{\bar{\gamma}}_t^{(i)} + \frac{\alpha^{(i)}\bar{G}_t^{(i)}(a)}{f_t^{(i)}(a)}\right) \end{aligned} \tag{12}$$

The sign of the first derivative of $f_t^{(i)}$ depends on the factor $\bar{\bar{\gamma}}_t^{(i)} + \frac{\alpha^{(i)}\bar{G}_t^{(i)}(a)}{f_t^{(i)}(a)}$ of expression (12) so it behaves as shown in (13).

$$\frac{\partial f_t^{(i)}}{\partial t}\begin{cases} <0 & \bar{\bar{\gamma}}_t^{(i)} + \frac{\alpha^{(i)}\bar{G}_t^{(i)}(a)}{f_t^{(i)}(a)} < 1 \\ =0 & \bar{\bar{\gamma}}_t^{(i)} + \frac{\alpha^{(i)}\bar{G}_t^{(i)}(a)}{f_t^{(i)}(a)} = 1 \\ >0 & \bar{\bar{\gamma}}_t^{(i)} + \frac{\alpha^{(i)}\bar{G}_t^{(i)}(a)}{f_t^{(i)}(a)} > 1 \end{cases} \tag{13}$$

Notice $\bar{\bar{\gamma}}_t^{(i)}$ is a constant for all $a \in A^{(i)}$ and $\int_{-\infty}^{+\infty} f_t^{(i)}(z)\,dz = 1$ so:

$$\begin{aligned} &\exists_{b_1^{(i)}, b_2^{(i)} \in A^{(i)}} : \frac{\bar{G}_t^{(i)}\left(b_1^{(i)}\right)}{f_t^{(i)}\left(b_1^{(i)}\right)} \neq \frac{\bar{G}_t^{(i)}\left(b_2^{(i)}\right)}{f_t^{(i)}\left(b_2^{(i)}\right)} \Longrightarrow \\ &\exists_{A_+^{(i)}, A_-^{(i)} \subset A^{(i)}, A_+^{(i)} \cap A_-^{(i)} = \emptyset}, \forall_{a_+^{(i)} \in A_+^{(i)}, a_-^{(i)} \in A_-^{(i)}} : \\ &\left(\frac{\partial f_t^{(i)}\left(a_+^{(i)}\right)}{\partial t} > 0\right) \wedge \left(\frac{\partial f_t^{(i)}\left(a_-^{(i)}\right)}{\partial t} < 0\right) \end{aligned} \tag{14}$$

From logical implication (14) it can be assured that the sign of $\frac{\partial f_t^{(i)}(a)}{\partial t}$ will be determined by the ratio $\frac{\bar{G}_t^{(i)}(a)}{f_t^{(i)}(a)}$. Notice subsets $A_+^{(i)}$ and $A_-^{(i)}$ are composed by the elements of $A^{(i)}$ that have not reached their value for the probability density function in equilibrium with $\bar{G}_t^{(i)}(a)$. That is, the $A_+^{(i)}$ subset is composed by all $a \in A^{(i)}$ having a value of probability density function which is too small with respect to $\bar{G}_t^{(i)}(a)$ and vice versa for $A_-^{(i)}$.

Let $a_*^{(i)} \in A^{(i)}$ be the actions yielding the highest value for every agent i at expression $\int_{-\infty}^{+\infty} \bar{\beta}_t^{(i)}(z)\, e^{-\frac{1}{2}\left(\frac{a-z}{\lambda_t^{(i)}}\right)^2} \mathrm{d}z$ as shown in (15). It is important to stress that $a_*^{(i)}$ are not the optimum of $\bar{\beta}_t^{(i)}$ given the set $f_t^{(j)}$ but the points yielding the optimal vicinity around it and defined by $e^{-\frac{1}{2}\left(\frac{a_*^{(i)}-z}{\lambda_t^{(i)}}\right)^2}$ which depends on $\lambda_t^{(i)}$.

$$\forall_{a \in A^{(i)}} : \int_{-\infty}^{+\infty} \bar{\beta}_t^{(i)}(z)\, e^{-\frac{1}{2}\left(\frac{a_*^{(i)}-z}{\lambda_t^{(i)}}\right)^2} \mathrm{d}z \geq \int_{-\infty}^{+\infty} \bar{\beta}_t^{(i)}(z)\, e^{-\frac{1}{2}\left(\frac{a-z}{\lambda_t^{(i)}}\right)^2} \mathrm{d}z \tag{15}$$

For symmetric games, that is common interest games ($\forall_{i,j} : \beta_t^{(i)} = \beta_t^{(j)}$), all $a_*^{(i)}$ will correspond initially to the same maximum in the joint-action space, if all learner start with the same parameters and an uniform PDF each one. Then (16) holds.

$$\begin{aligned}
\forall_t : f_t^{(i)}\left(a_*^{(i)}\right) \geq f_t^{(i)}\left(a^{(i)}\right) \Rightarrow f_{t+1}^{(i)}\left(a_*^{(i)}\right) \geq f_{t+1}^{(i)}\left(a^{(i)}\right) \\
\forall_t : \bar{\beta}_t^{(i)}\left(a_*^{(i)}\right) \geq \bar{\beta}_t^{(i)}\left(a^{(i)}\right) \Rightarrow \bar{\beta}_{t+1}^{(i)}\left(a_*^{(i)}\right) \geq \bar{\beta}_{t+1}^{(i)}\left(a^{(i)}\right) \\
\forall_t : \bar{G}_t^{(i)}\left(a_*^{(i)}\right) \geq \bar{G}_t^{(i)}\left(a^{(i)}\right) \Rightarrow \bar{G}_{t+1}^{(i)}\left(a_*^{(i)}\right) \geq \bar{G}_{t+1}^{(i)}\left(a^{(i)}\right)
\end{aligned} \tag{16}$$

Since the first derivative of the PDF depends on $\frac{\bar{G}_t^{(i)}(a)}{f_t^{(i)}(a)}$ the value necessary for keeping $\frac{\partial f_t^{(i)}(a)}{\partial t} = 0$ is also the highest for every $a_*^{(i)}$.

Notice that the maximum update that $f_t^{(i)}\left(a_*^{(i)}\right)$ may receive is obtained when $a_t^{(i)} = a_*^{(i)}$ – centering the bell at $a_*^{(i)}$ –, then if $f_t^{(i)}\left(a_*^{(i)}\right)$ reaches the value $\frac{1}{\lambda_t^{(i)}\sqrt{2\pi}}$, its first derivative will not be higher than 0 as shows (17) since $\beta : \Re \rightarrow [0,1]$.

$$\begin{aligned}
f_{t+1}^{(i)}\left(a_*^{(i)}\right) &= \bar{\gamma}_t^{(i)}\left(a_t^{(i)}\right) \\
&\left(f_*^{(i)}\left(a_t^{(i)}\right) + \beta_t^{(i)}\left(a_t^{(i)}\right)\alpha^{(i)} e^{-\frac{1}{2}\left(\frac{a_*^{(i)} - a_t^{(i)}}{\lambda_t^{(i)}}\right)^2}\right) \\
&\leq \frac{1}{1+\alpha^{(i)}\lambda_t^{(i)}\sqrt{2\pi}}\left(\frac{1}{\lambda_t^{(i)}\sqrt{2\pi}} + \alpha^{(i)}\right) \\
&\leq \frac{1}{1+\alpha^{(i)}\lambda_t^{(i)}\sqrt{2\pi}}\left(\frac{1+\alpha^{(i)}\lambda_t^{(i)}\sqrt{2\pi}}{\lambda_t^{(i)}\sqrt{2\pi}}\right) \\
&\leq \frac{1}{\lambda_t^{(i)}\sqrt{2\pi}}
\end{aligned} \tag{17}$$

Then the equilibrium point of $f_t^{(i)}\left(a_*^{(i)}\right)$ has the higher bound $\frac{1}{\lambda_t^{(i)}\sqrt{2\pi}}$. Notice that the closer the $\beta_t^{(i)}\left(a_*^{(i)}\right)$ to 1, the closer the equilibrium point of $f_t^{(i)}\left(a_*^{(i)}\right)$ to its higher bound.

$$\frac{\partial f_t^{(i)}\left(a_*^{(i)}\right)}{\partial t}\begin{cases} < 0 & f_t^{(i)}\left(a_*^{(i)}\right) > \frac{1}{\lambda_t^{(i)}\sqrt{2\pi}} \\ = 0 & f_t^{(i)}\left(a_*^{(i)}\right) = \frac{1}{\lambda_t^{(i)}\sqrt{2\pi}} \\ > 0 & f_t^{(i)}\left(a_*^{(i)}\right) < \frac{1}{\lambda_t^{(i)}\sqrt{2\pi}} \end{cases} \tag{18}$$

We can conclude from (18) that the highest value for $f^{(i)}$ will be achieved at $a_*^{(i)}$ as shown in (19) which has the higher bound $\frac{1}{\lambda_t^{(i)}\sqrt{2\pi}}$.

$$\begin{aligned}
\forall_{a \in A^{(i)} \setminus \{a_*^{(i)}\}} : lim_{t\rightarrow\infty} f_t^{(i)}(a) < f_t^{(i)}\left(a_*^{(i)}\right) \\
lim_{t\rightarrow\infty} f_t^{(i)}\left(a_*^{(i)}\right) \leq \frac{1}{\lambda_t^{(i)}\sqrt{2\pi}}
\end{aligned} \tag{19}$$

Finally

$$\begin{aligned}
lim_{\lambda^{(i)}\downarrow 0} lim_{t\rightarrow\infty} f_t^{(i)}\left(a_*^{(i)}\right) = \infty \\
\forall_{a \neq a_*^{(i)}} : lim_{\lambda^{(i)}\downarrow 0} lim_{t\rightarrow\infty} f_t^{(i)}(a) = 0
\end{aligned} \tag{20}$$

We can safety conclude that the set of CARLA will converge to an equilibrium in the joint-action space as long as the parameter α is low enough for the learners to match their expected policies through time. Which of the multiple equilibria is selected will depend on the initialization of the parameter λ.

Appendix B Analytical expression of rewards

$$
\begin{aligned}
\beta\left(a^{(1)},a^{(2)}\right) &= \text{Bell}\left(a^{(1)},a^{(2)},0.5,0.5,0.084\right) \\
&\bigcup 0.63\,\text{Bell}\left(a^{(1)},a^{(2)},0.1,0.9,0.187\right) \\
&\bigcup 0.63\,\text{Bell}\left(a^{(1)},a^{(2)},0.9,0.1,0.187\right) \\
\beta^{i}\left(a_t^{(1)},a_t^{(2)}\right) &= 0.3125\left(1+\sin(9a_t^{(1)}a_t^{(2)})\right)^2\frac{a_t^{(1)}+a_t^{(2)}}{2} \\
\beta^{ii}\left(a_t^{(1)},a_t^{(2)}\right) &= \text{Bell}\left(\left(a_t^{(1)}\right)^2,a_t^{(2)},0.5,0.5,0.002\right) \\
&\bigcup 0.7\,\text{Bell}\left(\left(a_t^{(1)}\right)^2,a_t^{(2)},0.1,0.1,0.2\right) \\
&\bigcup 0.7\,\text{Bell}\left(\left(a_t^{(1)}\right)^2,a_t^{(2)},0.9,0.9,0.2\right) \\
&\bigcup 0.7\,\text{Bell}\left(\left(a_t^{(1)}\right)^2,a_t^{(2)},0.9,0.1,0.2\right) \\
&\bigcup 0.7\,\text{Bell}\left(\left(a_t^{(1)}\right)^2,a_t^{(2)},0.1,0.9,0.2\right)
\end{aligned}
\tag{21}
$$

$$
\begin{aligned}
\beta_1(a) &= \bigcup_{i=1}^{n} f_i\, b_i(a) \\
\beta_2(a) &= \bigcup_{i=1}^{n} \left(r_i\, f_i\, b_i(a) + \text{rand}\,(1-r_i)\right) \\
\beta_3(a) &= pick\left\{\frac{r_i}{\sum_1^n r_i}, f_i\, b_i(a)\right\} \\
\beta_4\left(a^{(1)},a^{(2)}\right) &= \bigcup_{i=1}^{n} f_i \text{Bell}_i\left(a^{(1)},a^{(2)}\right)
\end{aligned}
$$

$$
b(a) = e^{-\frac{(a-\mu)^2}{2\sigma^2}}
$$

$$
\text{Bell}\left(a^{(1)},a^{(2)},\mu^{(1)},\mu^{(2)},\sigma\right) = e^{-\frac{\left(a^{(1)}-\mu^{(1)}\right)^2+\left(a^{(2)}-\mu^{(2)}\right)^2}{2\sigma^2}}
$$

$$
a \bigcup b = a + b - ab
$$