An RL Approach to Common-Interest Continuous Action Games

(Extended Abstract)

Abdel Rodríguez[†‡]
abrodrig@vub.ac.be

Peter Vrancx[†]
pvrancx@vub.ac.be

Ricardo Grau[‡]
rgrau@uclv.edu.cu

Ann Nowé[†]
ann.nowe@vub.ac.be

[‡]Center of Studies in Informatics
Universidad Central "Marta Abreu" de Las Villas
Santa Clara, Villa Clara, Cuba

[†]Computational Modeling Lab.
Vrije Universiteit Brussel
Plainlaan 2 1050 Brussels,
Belgium

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms

Keywords

Multi-agent learning, Teamwork, coalition formation and coordination, Implicit Cooperation

1. ABSTRACT

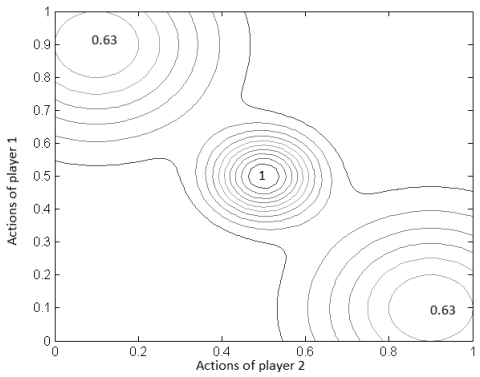
In this paper we present a reinforcement learning technique based on Learning Automata (LA), more specific Continuous Action Reinforcement Learning Automaton (CARLA), introduced by Howell et. al. in [2]. LA are policy iterators, which have shown good convergence results in discrete action games with independent learners. The approach presented in this paper allows LA to deal with continuous action spaces.

Recently, Rodríguez et al. [3] performed an analysis of the CARLA algorithm. The result of this analysis was 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 still has suboptimal performance with respect to global convergence in multi-agent settings.

The CARLA algorithm has successfully been applied to control problems [2, 1]. However in real world applications systems can be coupled and each subsystem is to be controlled by an individual controller. The interaction of these controllers can be considered as a common interest game. The interacting dynamics will have the learners converging to a suboptimal solution if the subsystems are controlled ignoring the existence of each other. In such a situation a better exploration of the joint-action space is required.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Figure 1: A two players game with three local optima. The contours represents combination of actions with the same reward.



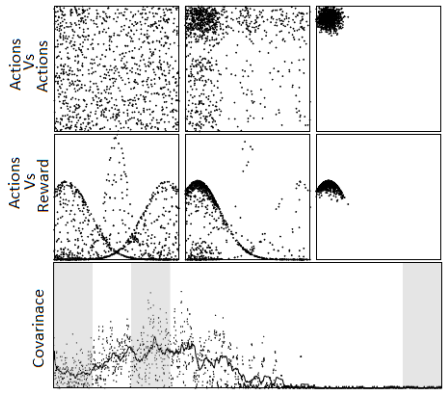
Exploring Selfish Reinforcement Learning (ESRL), introduced by [4], is an exploration method for independent LA playing a repeated discrete action game guaranteing convergence to the optimal Nash equilibrium. The supporting idea of this method is that a set of independent LA will converge to one of the Nash equilibria of the game, but not necessarily one from the Pareto front. ESRL proposes that once the agents converge to a Nash equilibrium, at least two learners should delete the selected action from their action spaces and restart learning. This allows the agents to find all dominant equilibria and agree on the best one. As the more interesting Nash equilibria are often also stronger attractors, the agents can quite efficiently reach Pareto optimal Nash equilibria.

This paper introduces Exploring Selfish Continuous Action Reinforcement Learning Automaton (ESCARLA), an extension of the ESRL method to continuous action games.

The supporting idea of ESRL is to exclude actions after every exploration phase. The problem with applying this approach in continuous action games, is that it makes no sense for the agents to delete a single action. Instead, a vicinity around the action should be identified and excluded. Now the agent must estimate when it crosses the boundary of the basin of attraction of the local attractor.

In order to solve this problem we propose to use the absolute

Figure 2: Relation between covariance and exploration.



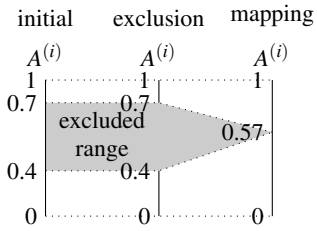
value of the covariance between the actions and rewards as a metric. Figure 1 shows the contour representation of a 2-players game example. There are three attractors in this example. The two local maxima located in the top left and bottom right corners have larger basins of attraction while the global maximum at the center has a narrower basin of attraction. Figure 2 shows the relation between the exploration and the covariance between actions and rewards from a single agent point of view. The first row shows a global view of the exploration. Three time intervals are shown. The first interval is the start of learning process (time-steps from 0 to 1000). The second interval is when the learners are reducing the global exploration (time-steps from 2000 to 3000). Notice this is a good time for deciding on which neighborhood to exclude. The last interval selected is when agents have converged to the local attractor (time-steps from 9000 to 10000). The second row shows the local information that the independent agents can access. The same time-steps are represented on each column but in this case we are plotting the selected actions on the horizontal axis and 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 time-steps corresponding to the three moments introduced above are shaded in gray. This covariance reaches a low value at the beginning of learning since lots of explorations are performed by both agents. When the agents are exploring within the basin of attraction of a local attractor then the noise in the rewards observed by each agent is minimal so the covariance reaches its maximum. As agents converge to a locally superior strategy, less exploration is performed so therefore the covariance value drops down to zero. The safe 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 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 (1) where $\text{percentile}(p,f)$ represents the value where the probability density function f accumulates the probability p.

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

The proposal here is to let the agents start learning until they all converge. Then each agent should delete the region defined by (1) from its action space. Deleting the action range implies modifying

Figure 3: Mapping process. Actions from the non-deleted $\{[0,0.4],[0.7,1]\}$ range will be mapped into the original action space



the action space so we need to map the new one into a compact set again as shown in Figure 3. Then all agents must restart the learning process to converge to another attractor. After enough exploration the agents should compare all the results and pick up the strategy that gave them the highest score. In engineering applications we may either know the total amount of maxima of the problem or the desired performance. In such applications the agents could understand by enough exploration by finding all different maxima of the problem or by achieving the desired performance. Please note that we are assuming a common interest game, so therefore the agents can agree on a best combination of actions. The general algorithm is given next.

```
ESCARLA algorithm
    repeat
        explore
        synchronize
    until enough exploration
    select best strategy
```

```
Exploration phase
    initialize parameters
    repeat
        sample action
        update strategy
        if maximum covariance
        then mark interval
    until convergence
```

```
Synchronization phase
    exclude marked interval
```

2. REFERENCES

[1] 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.

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

[3] 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.

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