# Learning to Cooperate in Multi-Agent Social Dilemmas

Enrique Munoz de Cote
Politecnico di Milano
Department of Electronics and Information
piazza Leonardo da Vinci 32, I-20133 Milan, Italy
munoz@elet.polimi.it

Alessandro Lazaric
Politecnico di Milano
Department of Electronics and Information
piazza Leonardo da Vinci 32, I-20133 Milan, Italy
lazaric@elet.polimi.it

Marcello Restelli
Politecnico di Milano
Department of Electronics and Information
piazza Leonardo da Vinci 32, I-20133 Milan, Italy
restelli@elet.polimi.it

## ABSTRACT

In many Multi-Agent Systems (MAS), self-interested agents need to cooperate in order to maximize their own utilities in time. The goal of this work is to improve cooperation among agents that use best-response Reinforcement Learning (RL) algorithms (Q-Learning), by the introduction of two new principles (*Change or Learn Fast* and *Change and Keep*) that foster the reaching of Pareto efficient stable outcomes.

## Categories and Subject Descriptors

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

## 1. INTRODUCTION

In this paper we focus on problems of cooperation among *self-interested agents*, where no explicit communication is possible and agents can only perceive the actions taken by the other agents at the previous time step.

Several algorithms [5, 2, 3] have been proposed to make agents able to learn Pareto efficient (PE) solutions.

In Section 2, we introduce two new principles to foster cooperation. Section 3 presents comparative results in a multi-agent social dilemma and Section 4 contains conclusions and future research directions.

## 2. MAKING Q-LEARNING COOPERATIVE

A matrix game is a tuple $\langle \mathcal{N}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, \{R_i\}_{i\in\mathcal{N}} \rangle$, where $\mathcal{N}$ is a collection of $n$ agents, $\mathcal{A}_i$ is the set of actions available to agent $i$, and $R_i$ is its payoff matrix. At each iteration $t$, let $\mathbf{a}^t = [a_1^t, a_2^t, ..., a_n^t]$ be the executed joint action, $a_i^t$ the action taken by the $i$-th agent and $\mathbf{a}_{-i}^t$ the joint action of all the players but $i$, $r_i^t$ is the reward received by $i$-th agent.

In this paper, we adopt a version of Q-learning where states are represented by the previous joint action ($\mathbf{a}^{t-1}$).

The desired outcome of our algorithms is to let a MAS of Q-learners collectively learn equilibrium points that payoff dominate the best response dynamics of normal Q-learners.


*AAMAS'06* May 8–12 2006, Hakodate, Hokkaido, Japan.


The main problem of pairing best-response agents in a repeated game is that they may end up in cyclic or suboptimal behaviors. In fact, many parallel learning processes may cause the environment to be non-stationary, thus preventing agents from predicting the correct outcome of their actions. In the following we introduce two variants of the Q-learning algorithm, aimed at improving cooperation of self-interested non-communicating Q-learning agents by taking into account the non-stationarity induced by the learning processes.

### 2.1 CoLF Principle

**Algorithm 1** COLF – Change Or Learn Fast

Let $\alpha_S > \alpha_{NS}$, and $\lambda$ be learning rates
$P(\mathbf{a}, a_i) \leftarrow S(\mathbf{a}, a_i) \leftarrow 0, Q(\mathbf{a}, a_i) \leftarrow \frac{r_{max}}{1-\gamma} \forall \mathbf{a} \in \mathcal{A}, a_i \in \mathcal{A}_i$
choose a random action $a_i^0$
execute $a_i^0$
read the joint action $\mathbf{a}^0$
$t \leftarrow 1$
**for all** steps **do**
  choose action $a_i^t$ according to exploration strategy
  execute $a_i^t$ and get the payoff $r_i^t$
  read the joint action $\mathbf{a}^t$
  $\Delta r_i^t \leftarrow |r_i^t - P(\mathbf{a}^{t-1}, a_i^t)|$
  **if** $\Delta r_i^t > S(\mathbf{a}^{t-1}, a_i^t)$ **then**
    $\alpha \leftarrow \alpha_{NS}$
  **else**
    $\alpha \leftarrow \alpha_S$
  **end if**
  $Q(\mathbf{a}^{t-1}, a_i^t) \leftarrow (1-\alpha) Q(\mathbf{a}^{t-1}, a_i^t) + \alpha \cdot \left(r_i^t + \gamma \cdot \max_{a_i} Q(\mathbf{a}^t, a_i)\right)$
  $S(\mathbf{a}^{t-1}, a_i^t) \leftarrow (1-\lambda) S(\mathbf{a}^{t-1}, a_i^t) + \lambda \cdot \Delta r_i^t$
  $P(\mathbf{a}^{t-1}, a_i^t) \leftarrow (1-\lambda) P(\mathbf{a}^{t-1}, a_i^t) + \lambda \cdot r_i^t$
  $t \leftarrow t+1$
**end for**

The CoLF (Change or Learn Fast) principle is inspired by the work of Bowling and Veloso [1], where a variable learning rate is considered. To foster cooperation, we propose to modify the learning rate of a Q-learning algorithm according to the following rule: if the payoff achieved by an agent is unexpectedly changing, then learn slowly, otherwise learn quickly. This principle aids in cooperation by giving less importance to "unexpected" payoffs, while allowing to speed up learning when the most of the agents are playing near-stationary strategies.

Algorithm 1 summarizes how the Q-learning algorithm changes with the introduction of the CoLF principle. The P-values are exponential averages of the collected payoffs with weight factor $\lambda$, while the S-values are exponential averages of the absolute differences between the current payoff and the respective P-value.

### 2.2 Change&Keep Principle

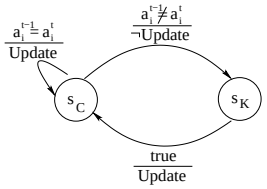



**Figure 1: CK finite-state machine. For each state transition are reported both the trigger condition (above the line) and whether the update phase occurs or not (below the line). In state $s_C$ the agent performs the usual action selection, while in state $s_K$ it repeats the previous action.**

The Change&Keep (CK) principle is based on the following observation: when an agent, due to either learning or exploration, decides to choose a different action, it typically collects an uninformative payoff. In fact, these (non-stationary) changes cannot be foreseen by other agents, and the related payoffs may be misleading, thus negatively affecting cooperation.

The idea of the CK principle is to discard the payoff received in correspondence to a change in the action selection (thus suspending the Q-value update), repeat the same action, and use the corresponding payoff for performing the suspended update. In this way the agent gives time for the other agents to react to its new action, thus using a more informative payoff for the update of its Q-table. Fig. 1 illustrates the two-states finite-state machine that describes the CK principle.

## 3. EXPERIMENTAL RESULTS

In this section, we compare the performance (in self-play) of Q-learning with those obtained by our variants (CK, CoLF, and CK-CoLF, which is obtained by combining both CK and CoLF principles together) in the MASD game [5]. In MASD, $N$ agents hold $M$ resource units each. At each iteration, let $a_i \in [0, M]$ be the amount of units contributed by agent $i$ towards a group goal $G$. The utility of agent $i$ given the joint action $\mathbf{a}$ is $P_i(\mathbf{a}) = \frac{\left[\frac{1}{N}\sum_{j=1}^{N} a_j\right] - k a_i}{M(1-k)}$, where $k \in \left(\frac{1}{N}; 1\right)$ is a constant that indicates how much each agent estimates its contribution towards the selfish goal. The payoff function is such that when all the agents put $M$ units in the group goal, each agent is rewarded with 1 and when nobody puts units in the group goal, a payoff of 0 is produced. If each agent adopts a random strategy the expected average payoff is 0.5. All the algorithms used an $\epsilon$-greedy exploration (where $\epsilon = \max(0.2 - 0.00006t, 0)$) joined with a relaxation search (as proposed in [5]) obtained by setting the initial values of the Q-table to high values.

In the learning rate performance comparison we used the following parametrization: $\gamma = 0.95$, and, for the CoLF principle, $\lambda = 0.1$. In Fig. 2 we report the results with different learning rates. The use of low learning rates in Q-learning (see Fig. 2(a)) allows to get higher payoffs but, as a drawback, the time required to reach a cooperative solution considerably increases. The results of CK (Fig. 2(b)) with the same learning rates show that it converges almost in the same time as Q-learning but the learning curve is steeper and outperforms Q-learning for every $\alpha$. In CoLF, we adopted $\alpha_{NS} = 0.1$ and $\alpha_S = 4 \cdot \alpha_{NS}$. As can be noticed in Fig. 2(c), the CoLF principle makes agents able to achieve higher payoffs than those obtained by Q-learning with constant learning rate. The algorithm succeeded in exploiting both the low learning rate in terms of convergence payoff performance and the high learning rate in terms of learning speed. Finally, Fig. 2(d) show how the introduction of the CoLF principle into the CK algorithm, improves the learning speed, while still reaching the PE solution.

We have experimentally verified (see Fig. 3) that high discount factors allow to achieve better performances, but with a lower learning speed. In fact, using 0.8 as discount factor, Q-learning and CoLF are not able to achieve a cooperative solution, which is obtained by adding the CK principle. On the other hand, all the algorithms achieve a cooperative solution, but they require much longer time to converge. We have also tested the performances of the four algorithms in different versions of the MASD problem obtained by varying the $k$ factor. For higher values of $k$ the agents become more selfish and the reaching of the cooperative solution is more difficult. The sensitivity analysis in Fig. 4 shows the performances of the four algorithms with different values of $k$. None of the four algorithms is able to cooperate with $k = 0.9$ since the selfish behavior is highly fostered. The problem becomes easier and easier as $k$ decreases.

## 4. DISCUSSION

As shown by the previous experiments, the CoLF principle is mainly concerned with non-stationarity caused by the other agents. It learns slow when gathering unexpected payoffs and fast otherwise. On the other hand, the CK principle deals with non-stationarity induced in the MAS by the behavior of the agent itself. Since actions selected following a non-stationary learning policy may not be foreseen by other agents, whenever an agent changes its action, according to the CK principle, it does not perform the update and repeats the same action and updates the related value with the last received payoff. The experiments carried out in the MASD framework show that the proposed principles applied to a best-response learning algorithm (Q-learning) largely improve its cooperation capabilities in self-play, both in terms of performance and learning speed.

## 5. ADDITIONAL AUTHORS

Additional authors: Andrea Bonarini (Politecnico di Milano, email: `bonarini@elet.polimi.it`).

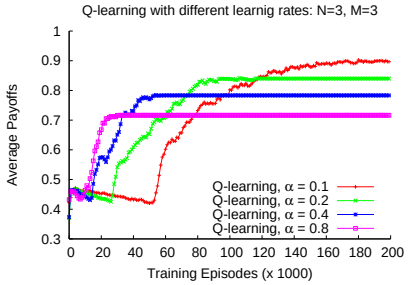
## 6. REFERENCES

[1] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

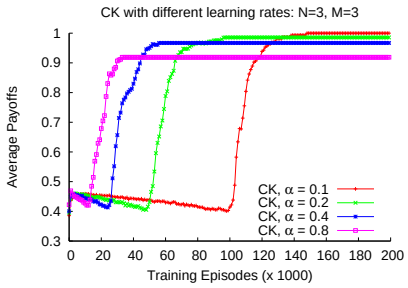
[2] J. W. Crandall and M. A. Goodrich. Learning to compete, compromise, and cooperate in repeated general-sum games. In *Proc. of ICML 2005*, to appear.

[3] M. W. Macy and A. Flache. Learning dynamics in social dilemmas. *Proceedings of the National Academy of Sciences*, 72(9):29–36, 2002.

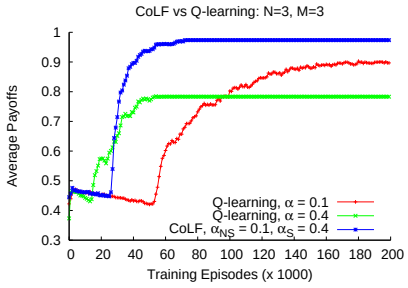
[4] J. L. Stimpson and M. A. Goodrich. Learning to cooperate in a social dilemma: A satisficing approach to bargaining. In *Proceedings of ICML*, 2003.



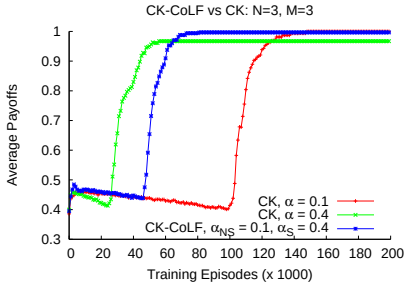

(a)




(b)

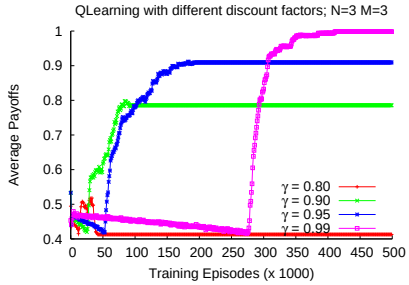



(c)

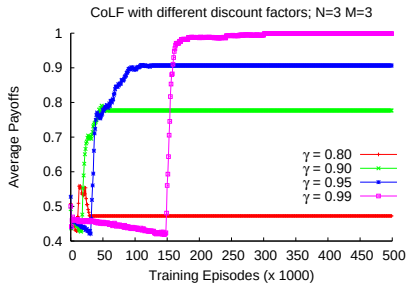



(d)

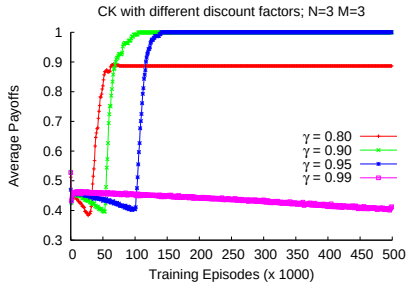
**Figure 2: Plots showing the effect of different learning rates in the MASD problem with 3 agents, 4 actions, and $k = 2/3$.**



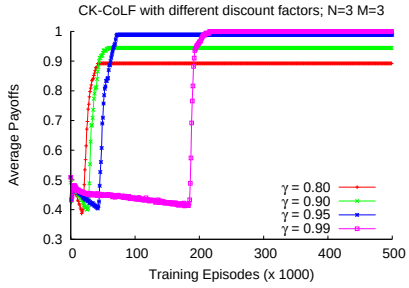

(a)




(b)

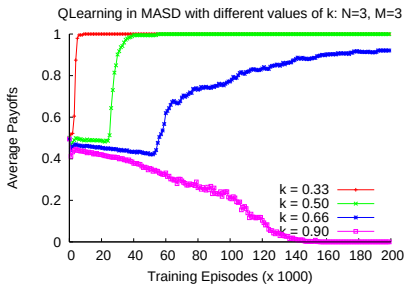



(c)

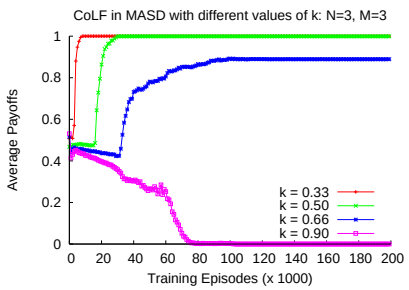



(d)

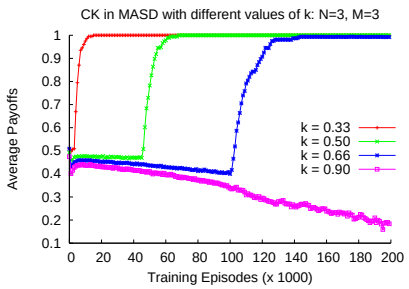
**Figure 3: Plots showing the effect of different discount factors in the MASD problem with 3 agents, 4 actions, and $k = 2/3$.**



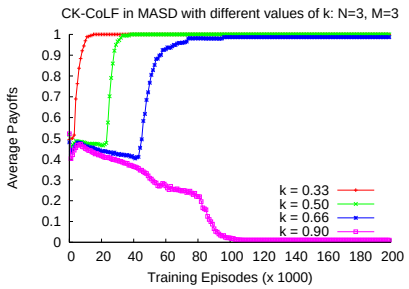

(a)




(b)




(c)




(d)

**Figure 4: Plots showing the effect of different values of $k$ in the MASD problem with 3 agents, 4 actions.**

[5] J. L. Stimpson, M. A. Goodrich, and L. C. Walters. Satisficing and learning cooperation in the prisoner's dilemma. In *Proc. of IJCAI 2001*, 2001.