A study of FMQ heuristic in cooperative multi-agent games.

Laëtitia Matignon, Guillaume J. Laurent, Nadine Le Fort - Piat

A study of FMQ heuristic in cooperative multi-agent games

Laëtitia Matignon, Guillaume J. Laurent and Nadine Le Fort-Piat

FEMTO-ST Institute, UMR CNRS 6174 - UFC / ENSMM / UTBM

Automatic Control and Micro-Mechatronic Systems Department

24, rue Alain Savary, 25000 Besançon, France.

E-mails : (laetitia.matignon,guillaume.laurent,nadine.piat)@ens2m.fr

Web site : www.femto-st.fr

Abstract

The article focuses on decentralized reinforcement learning (RL) in cooperative multi-agent games, where a team of independent learning agents (ILs) try to coordinate their individual actions to reach an optimal joint action. Within this framework, some algorithms based on Q-learning are proposed in recent works. Especially, we are interested in Distributed Q-learning which finds optimal policies in deterministic games, and in the Frequency Maximum Q value (FMQ) heuristic which is able in partially stochastic matrix games to distinguish if a poor reward received for the same action are due to either miscoordination or to the noisy reward function. Making this distinction is one of the main difficulties to solve stochastic games. Our objective is to find an algorithm able to switch over the updates according to a detection of the cause of noise. In this paper, a modified version of the FMQ heuristic is proposed which achieves this detection and the update adaptation. Moreover, this modified FMQ version is more robust and very easy to set.

1 Introduction

In recent years there has been increased interest in decentralized approaches combined with artificial intelligence to solve complex real world problems. Machine learning is a popular approach to study multi-agent systems (MAS) and a decentralized point of view offers several potential advantages as speed-up, scalability and robustness [11]. We focus on learning in cooperative multi-agent systems [8], where multiple agents are cooperating to solve a joint task. Especially, we are interested in learning in MAS thanks to reinforcement learning (RL) methods, where an agent learns by interacting with its environment, using a scalar reward signal called reinforcement as performance feedback [12]. Over the last decade, many approaches are concerned with the extension of RL to MAS [2].

In this framework, Claus & Boutilier [3] distinguish two cases of reinforcement learners : the case of agents that get information about their own choice of action as well as their partners' choices, called "*joint action learners*" (JALs), and the case of agents which only know their own action, called "*independent learners*" (ILs). Even though it would be "easier" for a JAL to solve a joint task than with only individual actions, we focus on ILs which is a more realistic assumption and don't require any communication between agents. Moreover, given that ILs don't get information about all of the other agents' actions, they bring the benefit in Q-learning variants of a state-action space size independent of the number of agents. The main difficulty with ILs is the *multi-agent coordination problem* : how to make sure that all ILs coherently choose their individual action such that the resulting joint action is optimal ?

Over the last years, many algorithms addressing this problem of coordination in cooperative matrix games (MG) have been proposed. Some of them find optimal policies in deterministic cooperative MG where mis-coordination is associated with high penalties, *e.g.* the Distributed Q-learning [7]. However, they fail in stochastic MG where the fundamental difficulty is to *make distinctions between the noise induced by the stochastic reward function and by other agents' evolving behaviour.* Thus, it could be interesting to detect if the game is deterministic and then to use a fitted algorithm as the Distributed Q-learning, and otherwise to switch over the algorithm. In this perspective, we took an interest in the Frequency Maximum Q value (FMQ) heuristic [5, 6]. FMQ is able in partially stochastic MG to distinguish if the different rewards received for the same action are due to either the other agents or to the noisy reward function. This paper presents a study of some limitations and improvements of the FMQ.

Our paper is structured as follows: we first introduce the framework of fully cooperative repeated matrix games and common testbeds for the study of coordination of agents. We then review related works dealing with RL algorithms for ILs. Finally, we investigate some issues of the FMQ and suggest a modified FMQ more robust and easy to set.

2 Fully cooperative repeated matrix games

The studies of learning algorithms in MAS are based on game theory and more particularly on repeated games. In this section, we first setup this framework. Then, we present widely used testbeds for studying RL in repeated matrix games.

2.1 Definition

A *matrix game*[1] (MG) is a multiple-agent, single state framework. It is defined as a tuple $< m, A_1, ..., A_m, R_1, ..., R_m >$ where m is the number of players, A_i is the set of actions available to player i (and $A = A_1 \times ... \times A_m$ is the joint action space) and $R_i : A \mapsto \Re$ is player i's payoff function.

If $R_1 = ... = R_m = R$, the MG is *fully cooperative*[2]. We are interested in *repeated games* which consist of the repetition of the same MG by the same agents. Among matrix games, bi-matrix games are often used to represent cooperative coordination problems in the 2-agents case.

2.2 Cooperative matrix games

Table 1 shows four cooperative MG interesting for the study of the *coordination* of two agents. The Climbing game and the Penalty game have been introduced in [3] and partially and fully stochastic variations of the Climbing game, proposed in [5]. Each agent has 3 actions and the table specifies the joint rewards. Each of these games is challenging due to mis-coordination penalties. In the Climbing game, the optimal joint action is (a, a) but if an agent chooses its individual optimal action a when the other agent chooses action b, a severe penalty is received. However, there are no mis-coordination penalties associated with action c, potentially making it tempting for the agents. In the stochastic variations of the Climbing game, the agents confront the situation of a noisy reward function. The difficulty is to *distinguish if the poor rewards received for the same action are due to either mis-coordination or to the noisy reward function.* For instance, even if the highest reward of 14 is sometimes received when executing b, an agent

[1]also called *strategic game*
[2]also called *team game*

(a) Climbing game

		Agent 2		
		a	b	c
	a	11	-30	0
Agent 1	b	-30	7	6
	c	0	0	5

(b) Partially stochastic Climbing game

		Agent 2		
		a	b	c
	a	11	-30	0
Agent 1	b	-30	14/0	6
	c	0	0	5

(c) Fully stochastic Climbing game

		Agent 2		
		a	b	c
	a	10/12	5/-65	8/-8
Agent 1	b	5/-65	14/0	12/0
	c	8/-8	12/0	10/0

(d) Penalty game

		Agent 2		
		a	b	c
	a	10	0	k
Agent 1	b	0	2	0
	c	k	0	10

Table 1: Cooperative matrix games. In stochastic games, the probability of each reward is 50%.

must learn that the average reward for the joint action (b, b) is lower than that of (a, a).
In the Penalty game, k is usually chosen inferior to 0. This game introduces another mis-coordination issue due to the presence of two optimal joint actions (a, a) and (c, c): simply choosing its individual optimal action does not guarantee that the other agent will choose the same optimal.

3 Related works in Matrix Games

In this section, related works dealing with the coordination of agents thanks to RL algorithms in fully cooperative repeated games are reviewed, with an emphasis on research dealing with Q-learning [14] and Q-learning variants for ILs. In the following methods, each ILs builds its own Q-table whose size is independent of the agents number and linear in function of its own actions. The task is for the agents to independently choose one action with the goal of maximizing the reward that they receive.

3.1 Decentralized Q-learning

Q-learning [14] was one of the first algorithm applied to multi-agent environments [13]. In the framework of MG, Claus & Boutilier [3] reduce Q-learning's update equation to :

$$Q(a) \leftarrow (1-\alpha)Q(a) + \alpha r \tag{1}$$

where a is the agent's chosen action, r the reward received, $Q(a)$ the action-value and $\alpha \in]0;1]$ the learning rate.

The agent individual policy $\pi : A \mapsto [0,1]$ returns a probability distribution over individual actions. At time k,

$$\forall a \in A,\ P(\underline{a}_k = a) = \pi(a) \tag{2}$$

The policy π can be computed thanks to a Boltzman distribution :

$$\pi(a) = \frac{e^{\frac{Q(a)}{\tau}}}{\sum\limits_{u \in A} e^{\frac{Q(u)}{\tau}}} \tag{3}$$

where τ is the temperature parameter that decreases the amount of randomness as it approaches zero. This action decision is called the softmax strategy. The ϵ-greedy strategy is another common action selection method in which an agent chooses the best action according to its policy with probability $(1-\epsilon)$ (exploitation mode), and otherwise selects a uniformly random action with probability ϵ (exploration mode).

As for the decision strategy, Claus & Boutilier [3] use softmax strategy with a temperature parameter τ decreasing over time so that the exploitation probability increases. Notably, they try to compute an equilibrium point by continuously reducing the exploration frequency, so as to avoid *concurrent exploration*. Anyway, with such a strategy, the key difficulty is that convergence relies on the use of decaying exploration and so convergence to an optimal equilibrium with decentralized Q-learning is not ensured. Decaying the value of the temperature is also investigated in [4] and in §4.1.

3.2 Distributed Q-learning

Lauer & Riedmiller [7] introduce “optimistic independent agents”: they neglect in their update the penalties which are often due to a non-coordination of agents. Thus the evaluation of an action in MG is the maximum reward received. In the case of multiple optimal joint actions in a single state (for instance the Penalty game), an additional procedure for coordination is used. The central idea is to update the current policy π only if an improvement in the evaluation values (R_{max}) happens. Distributed Q-learning associated with this coordination method is in Algorithm 1. It is proved that this algorithm finds optimal policies in deterministic environments for cooperative multi-stage MG. Anyway this approach does not converge in stochastic environments.

Algorithm 1: Distributed Q-learning for Matrix Game for agent i

```
1  begin
2      Initialization : $\forall a \in A,\ R_{max}(a) \leftarrow 0,\ \pi(a)$ arbitrarily
3      repeat
4          Select $a$ according to the $\epsilon$-greedy action selection method
5          Apply $a$ and observe reward $r$
6          if $r > R_{max}(a)$ then
7              $R_{max}(a) \leftarrow r$
8              $\forall b \in A \quad \pi(b) \leftarrow \begin{cases} 1 & \text{if } a = b \\ 0 & \text{else} \end{cases}$
9      until stop
10 end
```

3.3 Hysteretic Q-learning

In [1], two learning rates α and β are used for the increase and decrease rates of Q-values in order to overcome the issue of coordination in MAS. The update equation in MG is :

$$Q(a) \leftarrow \begin{cases} (1-\alpha)Q(a) + \alpha r & \text{if } \ r \geq Q(a) \\ (1-\beta)Q(a) + \beta r & \text{else} \end{cases} \tag{4}$$

The idea is that agents should not be altogether blind to penalties at the risk of staying in sub-optimal equilibrium or mis-coordinating on the same optimal joint action. But they are chiefly optimistic to reduce oscillations in the learned policy ($\alpha > \beta$).

3.4 Lenient Learners

Panait *et al.* [9] are interested in varying the degree of optimism of the agents as the game is repeated. Indeed, being optimistic may be useful at early stages of learning to identify promising actions. In case of lenient learners, agents are exploring at the beginning so most of selected actions are poor choices and ignoring penalties is then justified. Nevertheless, it may lead to an overestimation of actions, especially in stochastic domains where rewards are noisy. And once agents have explored, it becomes interesting to achieve accurate estimation of actions. So the agents are initially lenient (or optimistic) and the degree of lenience concerning an action decreases as the action is often selected. The main drawbacks of this method is that a large number of parameters must be set.

Algorithm 2: FMQ for Matrix Game for agent i

```
begin
    Initialization : $\forall a \in A,\ Q(a) \leftarrow 0,\ R_{max}(a) \leftarrow 0, C(a) \leftarrow 0$
    $C_{R_{max}}(a) \leftarrow 0, F(a) \leftarrow 1, E(a) \leftarrow 0, \pi$ arbitrarily
    repeat
        Select $a$ following the policy $\pi$
        Apply $a$ and observe reward $r$
        $C(a) \leftarrow C(a) + 1$
        $Q(a) \leftarrow (1 - \alpha)Q(a) + \alpha r$
        if $r > R_{max}(a)$ then
            $R_{max}(a) \leftarrow r$
            $C_{R_{max}}(a) \leftarrow 1$
        else if $r = R_{max}(a)$ then
            $C_{R_{max}}(a) \leftarrow C_{R_{max}}(a) + 1$
        $F(a) \leftarrow \frac{C_{R_{max}}(a)}{C(a)}$
        $E(a) \leftarrow Q(a) + c \times F(a) \times R_{max}(a)$
        $\forall b \in A,\ \pi(b) \leftarrow \frac{e^{\frac{E(b)}{\tau}}}{\sum_{u \in A} e^{\frac{E(u)}{\tau}}}$
    until stop
end
```

3.5 FMQ

Kapetanakis & Kudenko [5, 6] bias the probability of choosing an action with the frequency of receiving the maximum reward for that action. In their algorithm, the evaluation of an action E is the Q-value added to an *heuristic value*, taking into account how often an action produces its maximum corresponding reward. The evaluation of an action a is defined as :

$$E(a) = Q(a) + c \times F(a) \times R_{max}(a) \tag{5}$$

where $R_{max}(a)$ is the maximum reward received so far for choosing action a, $F(a)$ is the frequency of receiving the maximum reward corresponding to an action, and c is a weight which controls the importance of the FMQ heuristic in the evaluation.
The Frequency Maximum Q value (FMQ) algorithm is the Algorithm 2. $C(a)$ holds the number of times the agent has chosen the action a in the game and $C_{R_{max}}(a)$ the number of times that the maximum reward has been received as a result of playing a.

	Decentralized Q-Learning	Distributed Q-Learning	Hysteretic Q-Learning	FMQ
Climbing game	12%	100%	100%	100%
Penalty game ($k = -100$)	64%	100%	100%	100%
Partially Stochastic Climbing game	-	7%	60%	100%
Fully Stochastic Climbing game	-	-	-	21%

Table 2: Percentage of trials which converged to the optimal joint action.

3.6 Conclusion

To conclude, we compare the performance of some of these algorithms in cooperative MG (Table 1). A trial consists of 5000 repetitions of the game. At the end of each trial, we determine if the greedy joint action is the optimal one. We performed 500 trials. Results were obtained with the best chosen action selection strategy in order to achieve the best results of convergence. Table 2 brings together the percentage of trials converging to the optimal joint action according to the algorithm and the type of cooperative MG.
All the algorithms, except the Decentralized Q-learning, converge in deterministic games. FMQ is the only one to overcome the difficulty of partially noisy rewards. Thus, *the advantage of the FMQ is to be able to make the distinction between the noise due to the stochastic rewards and the noise due to the other agent.* This heuristic is an interesting way to detect the stochasticity of a game. However, a large number of parameters must be set and convergence relies on the choice of these parameters. That's why FMQ improvements are presented in the next section.

4 A study of FMQ in cooperative MG

FMQ heuristic performs well in cooperative matrix games where the joint reward is partially noisy. However, there remain questions towards understanding exactly how the exploration strategy influences the convergence. The convergence also relies on the choice of the weight parameter. In this section, we investigate these issues in details and suggest modifications to the FMQ algorithm to overcome these limitations. *The objective is to make the algorithm more robust to the choice of the exploration strategy and to get rid of the choice of the weight parameter c.*

4.1 The issue of the exploration strategy

Decaying the exploration rate (ϵ or τ) in an appropriate manner is a popular choice in single-agent method, especially when using on-policy RL algorithms, such as SARSA, in order to ensure convergence to the optimal (deterministic) policy [10]. In practice, however, constant exploration rates are used in single-agent method.
In multi-agent method, Claus & Boutilier [3] decay the value of the temperature to avoid concurrent exploration, but this is without ensuring convergence to an optimal equilibrium (§3.1). For their part, Kapetanakis & Kudenko [5] use with their FMQ heuristic an exponentially decaying temperature function :

$$\tau_k = e^{-\delta k} \times \tau_{max} + \tau_\infty \tag{6}$$

where k is the number of repetitions of the game so far, δ controls the rate of the exponential decay, τ_{max} and τ_∞ are *resp.* the value of the temperature at the beginning and at the end of the trial. However, using such a temperature function requires to choose in an appropriate manner all the parameters. For instance, in the case of heterogeneous cooperative multi-agent systems [6], *changing just one setting in these parameters can turn a successful experiment into an unsuccessful one.* This instability is also more important in multi-stage games.

In Figure 1, an exponentially decaying temperature is plotted in function of the number of repetitions. The other curves are the average rewards and their dispersion received by FMQ agents using this temperature function and the softmax strategy in the Climbing game. During a first phase, the average rewards remain constant; the agents choose all possible joint actions. This is the phase of *exploration.* Then, during the exponential decay of τ, the agents learn to coordinate until the temperature reaches some lower limit where agents are following their greedy policy. This is the phase of convergence or *coordination.* Both of these phases of exploration and coordination are necessary for the FMQ to converge. For instance, with a constant exploration rate $\epsilon = 0.1$, the FMQ algorithm converges to the optimal joint action in the Climbing game with a probability of only 23%. So the FMQ algorithm is not robust to the choice of the exploration strategy.

This link between convergence and exploration is due to an instability of the frequency of receiving the maximum reward F face to the exploration. For instance in the Climbing game[3], the action a can be chosen many times

[3]an agent has 3 actions a, b and c

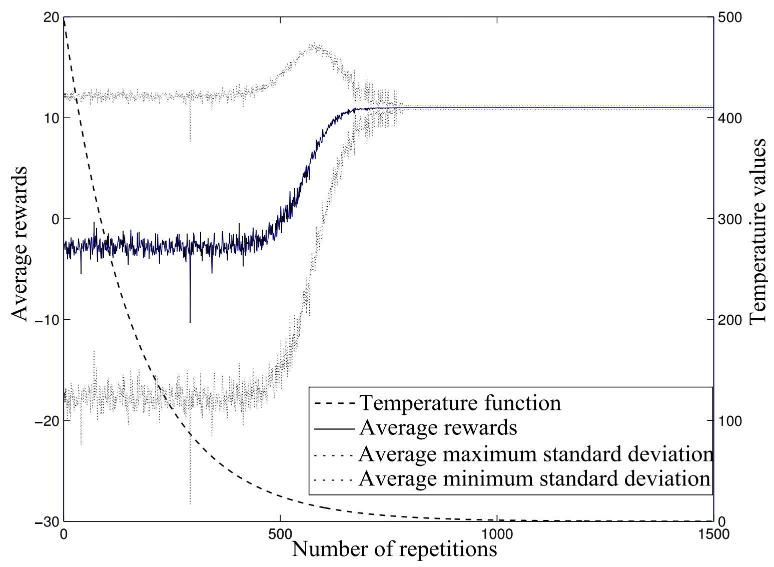

Figure 1: Average rewards received in the Climbing game by FMQ agents ($c = 10$) with $\tau = e^{0.006 \times k} \times 499 + 1$.

by an agent but without leading to its maximum reward. So $C(a)$ is high. Then, when the maximum reward is received for the action a (because of an exploration step), $C_{R_{max}}(a)$ is reset and thus, the frequency $F(a) \leftarrow \frac{1}{C(a)}$ drops. $F(a)$ might then be inferior to $F(b)$ or $F(c)$ and the action a might be chosen only in exploration step. So the convergence is not effective and many exploration steps are necessary so that the frequency $F(a)$ may increase enough for the coordination on the joint action.

In order to obtain an algorithm more robust to the exploration, the counter $C(a)$ must be reset to 1 when a new maximum reward is received for the action a. Thus, the frequency is also reset to 1 and will only decrease in case of mis-coordination because of an exploration step of one agent or in case of noisy reward. Besides, we introduce a recursive computation of the frequency:

$$F(a) \leftarrow \begin{cases} (1-\alpha_f)F(a) + \alpha_f & \text{if} \ \ r = R_{max}(a) \\ (1-\alpha_f)F(a) & \text{else} \end{cases} \tag{7}$$

where α_f is the learning rate of the frequency.

With this modified version of FMQ, *i.e.* a reset and a recursive computation of the frequency F and the ϵ-greedy action-selection method, FMQ agents converge with a probability of 100% to the optimal joint action in the deterministic and partially stochastic Climbing game (with parameters $\alpha = 0.1$, $\alpha_f = 0.05$, $c = 10$ and $\epsilon = 0.1$). *This version of FMQ with a constant exploration rate performs as well in the Climbing game as the initial version with an exponentially decaying temperature function.*

The improvement concerns the fact that *this modified version of FMQ is more robust to the exploration*, but it does not do any better than the original formulation. This modified algorithm converges with a well chosen decaying temperature and with a constant exploration rate. Anyway, the second choice requires less parameters to set and is *less inclined to instability due to the exploration.* The only care concerns a sufficient number of repetitions of the game according to the constant exploration rate to ensure complete exploration of the action space.

4.2 The issue of the weight in the FMQ heuristic

Kapetanakis & Kudenko [5] introduce a weight c to control the importance of the FMQ heuristic in the evaluation of an action (equation 5). But the convergence relies on the value of this parameter, as shows experiments performed in [5]. In order to overstep this issue, we propose another heuristic evaluation.

We notice that R_{max} is in most cases an overestimation of the evaluation of an action, except in deterministic games where it is the exact maximal value of an action. And Q is an estimation of the average rewards, including poor values due to mis-coordination. So the evaluation E must be equal to R_{max} when the game is deterministic, *i.e.* when the frequency of receiving the maximum reward corresponding to an action is equal to 1. Otherwise, E is inferior to R_{max}. We propose to use the following evaluation of action :

$$E(a) = (1 - F(a)) \times Q(a) + F(a) \times R_{max}(a). \tag{8}$$

When a new maximum reward for an action is received, the agent is then optimistic concerning this action. Then, if the reward is noisy, it becomes less optimistic and chooses its action according to Q-values. The principle is then closest in spirit to lenient learners [9] (§3.4), in that the degree of optimism of the agents can change. But concerning lenient learners, the degree of optimism inevitably decreases, although here, *if the agents manage to coordinate in a deterministic environment, they stay optimistic.* So they are bound to converge to the optimal joint action in deterministic environment, given that they follow the Distributed Q-learning. The additional procedure for coordination of Lauer & Riedmiller [7] is also used, with an update of the current policy π only if an improvement in the evaluation values (E) happens.

Results on cooperative matrix games with the modified version of FMQ (Algorithm 3) are in Table 3 ($\epsilon = 0.1$, $\alpha = 0.1$ and $\alpha_f = 0.05$). This modified version of FMQ with an on-line computation of the frequency of receiving the maximum reward and a constant exploration rate performs as well as the initial version in the Climbing game and in its partially stochastic version. But it does not overcome the issue of strongly noisy reward functions (only 35% of trials converge to the optimal joint action). However, we would not expect our modified FMQ version to have any positive impact on the convergence results in the fully stochastic version; *we would only expect to obtain a more robust algorithm.*

Algorithm 3: Modified FMQ for Matrix Game for agent i

```
1  begin
2      Initialization :
3      forall actions $a = 1..N$ do
4          $Q(a) \leftarrow 0$, $R_{max}(a) \leftarrow 0$, $F(a) \leftarrow 1$, $E(a) \leftarrow 0$, $\pi(a)$ arbitrarily
5      repeat
6          Select $a$ according to the $\epsilon$-greedy action selection method
7          Apply $a$ and observe reward $r$
8          $Q(a) \leftarrow Q(a)(1-\alpha) + \alpha r$
9          if $r > R_{max}(a)$ then
10             $R_{max}(a) \leftarrow r$
11             $F(a) \leftarrow 1$
12         else if $r = R_{max}(a)$ then
13             $F(a) \leftarrow (1-\alpha_f)F(a) + \alpha_f$
14         else
15             $F(a) \leftarrow (1-\alpha_f)F(a)$
16         $E(a) \leftarrow (1-F(a)) \times Q(a) + F(a) \times R_{max}(a)$
17         if $E(\arg\max_{u \in U} \pi(u)) \neq \max_{u \in U} E(u)$ then
18             Select a random action $a_{max} \in \arg\max_{u \in U} E(u)$
19             $\forall b \in A \;\; \pi(b) \leftarrow \begin{cases} 1 & \text{if } \; b = a_{max} \\ 0 & \text{else} \end{cases}$
20     until stop
21 end
```

	Decentralized Q-Learning	Distributed Q-Learning	Hysteretic Q-Learning	FMQ	Improved FMQ
Climbing game	12%	100%	100%	100%	100%
Penalty game ($k = -100$)	64%	100%	100%	100%	100%
Partially Stochastic Climbing game	-	7%	60%	100%	100%
Fully Stochastic Climbing game	-	-	-	21%	35%

Table 3: Percentage of trials which converged to the optimal joint action.

5 Conclusion

In this paper, we are interested in the coordination of agents in cooperative matrix games. In this framework, we reviewed some algorithms. Especially, we study FMQ heuristic whose main interest is its ability to distinguish if the different rewards received for the same action are due to either the other agent or to the partially noisy reward function. We have removed some limitations of the FMQ and proposed in this paper a modified FMQ algorithm. An on-line computation of the frequency of receiving the maximum reward is suggested as well as a novel evaluation of the heuristic. The main idea of this new heuristic is that the agent is initially optimistic and the degree of optimism will decrease in advantage of Decentralized Q-learning if the reward is noisy. The advantage is to be robust to the exploration and to have only three parameters very easy to set.

Changes made to the FMQ algorithm lead to the same convergence results than the original formulation, but it lives up to ours' expectations of a more robust algorithm face to the choice of the exploration strategy and to the choice of parameters. However, the modified FMQ version is only proposed for cooperative matrix games. So we are currently investigating an extension of this new version of FMQ to multi-stage games.

Acknowledgments

This work is partially supported by the Smart Surface ANR (French National Research Agency) project (ANR_06_ROBO_0009).

References

[1] Laetitia Matignon andGuillaume J. Laurent and Nadine Le Fort-Piat. Hysteretic q-learning :an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2007*, pages 64–69, San Diego, CA, USA, Nov. 2007.

[2] Lucian Busoniu, Robert Babuska, and Bart De Schutter. Multi-agent reinforcement learning: A survey. In *Int. Conf. Control, Automation, Robotics and Vision*, pages 527–532, December 2006.

[3] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, 1998.

[4] S. Kapetanakis and D. Kudenko. Improving on the reinforcement learning of coordination in cooperative multi-agent systems. Second Symposium on Adaptive Agents and Multi-Agent Systems(AISB/AAMAS-II), Imperial College, London, April 2002.

[5] S. Kapetanakis and D. Kudenko. Reinforcement learning of coordination in cooperative multi-agent systems. In *Proceedings of the Nineteenth NCAI*, 2002.

[6] S. Kapetanakis and D. Kudenko. Reinforcement learning of coordination in heterogeneous cooperative multi-agent systems. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1258–1259, Washington, DC, USA, 2004. IEEE Computer Society.

[7] 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.

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

[9] Liviu Panait, Keith Sullivan, and Sean Luke. Lenient learners in cooperative multiagent systems. In *AAMAS '06: Proceedings of the fifth inter-*

national joint conference on Autonomous agents and multiagent systems, pages 801–803, New York, NY, USA, 2006. ACM Press.

[10] Satinder P. Singh, Tommi Jaakkola, Michael L. Littman, and Csaba Szepesvari. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38(3):287–308, 2000.

[11] Peter Stone and Manuela M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000.

[12] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* The MIT Press, Cambridge, 1998.

[13] M. Tan. Multiagent reinforcement learning: Independent vs. cooperative agents. In *10th International Conference on Machine Learning*, pages 330–337, 1993.

[14] C.J.C.H. Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning*, 8:279–292, 1992.