Actor-Critic Fictitious Play in Simultaneous Move Multistage Games

Julien Perolat[1]
Univ. Lille

Bilal Piot[1]
Univ. Lille

Olivier Pietquin[1]
Univ. Lille

Abstract

Fictitious play is a game theoretic iterative procedure meant to learn an equilibrium in normal form games. However, this algorithm requires that each player has full knowledge of other players' strategies. Using an architecture inspired by actor-critic algorithms, we build a stochastic approximation of the fictitious play process. This procedure is on-line, decentralized (an agent has no information of others' strategies and rewards) and applies to multistage games (a generalization of normal form games). In addition, we prove convergence of our method towards a Nash equilibrium in both the cases of zero-sum two-player multistage games and cooperative multistage games. We also provide empirical evidence of the soundness of our approach on the game of Alesia with and without function approximation.

1 Introduction

Go, Chess, Checkers, Oshi-Zumo [10] are just a few example of Multistage games [7]. In these games, the interaction proceeds from stage to stage without looping back to a previously encountered situation. This model groups a broad class of multi-agent sequential decision processes where the interaction never goes back in the same state. This work focuses on Multi-Agent Reinforcement Learning [11] (MARL) in Multistage games. In this multi-agent environment, players evolve from state to state as a result of their mutual actions. During this interaction, all players receive a reward informing on how good was their action when performed in the state they were in. The goal of MARL is to learn a strategy that maximally accumulates rewards over time. Whilst the problem is fairly well understood when studying single agent RL, learning while independently interacting with other agents remains superficially explored. The range of open questions is so wide in that area [31] that it is worth giving a precise definition of our goal. In this paper, we follow a prescriptive agenda. We intend to find a learning algorithm that provably converges to Nash equilibrium in cooperative and in non-cooperative games. The goal is to find a strategy that can be executed independently by each player that corresponds to a Nash equilibrium. Many, if not most, approaches to address this problem consider a centralized learning procedure that produces an independent strategy for each player [21]. Centralized learning procedures are quite common and often perform better than decentralized learning procedures [13]. But these centralized learning procedure require synchronization between agents during learning (which is the main limitation of these methods). The agenda we follow in this paper is to propose a decentralized on-line learning method that provably converges to a Nash equilibrium in self-play. Decentralized algorithms, because they allow building identical independent agents that don't rely on anything but the observation of their state and reward, no central controller being required. On-line algorithms, on another hand, allow learning while playing and do not require prior computation of possible strategies.

This agenda is a fertile ground of interaction between traditional RL and game theory. Indeed, RL aims at building autonomous agents learning on-line in games against nature (where the environment in not interested in wining). For that reason, a wide variety of single agent RL algorithms have been adapted to multi-agent problems. But several major issues prevent direct use of standard RL with multi-agent systems. First, blindly applying single agent RL in a decentralized fashion implies that, from each agent's point of view, the other agents are part of the envi-

[1] now with DeepMind, London (UK)

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84. Copyright 2018 by the author(s).

ronment. Such an hypothesis breaks the crucial RL assumption that the environment is (at least almost) stationary [22]. Second, it introduces partial observability as each agent's knowledge is restricted to its own actions and rewards while its behavior should depend on others' strategies.

Decentralized procedures (unlike counterfactual regret minimization algorithms [34]) have been the topic of many studies in game theory and many approaches were proposed from policy hill climbing methods [8, 2] to evolutionary dynamics [33, 1] (related work will be detailed in Sec. 2). But those dynamics do not converge in all general-sum normal-form games, and, there exists a three-player normal form game [15] for which no first order uncoupled dynamics (*i.e.* most decentralized dynamics) can converge to a Nash equilibrium. Despite this counterexample, decentralize dynamics remain an important case to study because building a central controller for a multi-agent system is not always possible nor is observing the actions and rewards of every agent. Even if decentralized learning processes (as described in [15]) will never be guaranteed to converge in general, they should be at least guaranteed to converge in some interesting classes of games such as cooperative and zero-sum two-player games.

Fictitious play is a model-based process that learns Nash equilibria in normal form games. It has been widely studied and required assumptions were weakened over time [23, 17] since the original article of Robinson [28]. It has been extended to extensive form games (game trees) and, to a lesser extent, to function approximation [16]. However it is neither on-line nor decentralized except from the work of [23] which focuses on normal form games and [16] that has weak guarantees of convergence and focus on turn taking imperfect information games. Fictitious play enjoys several convergence guarantees [17] which makes it a good candidate for learning in simultaneous multistage stage games.

This paper contributes to fill a gap in the MARL literature by providing two online decentralized algorithms converging to a Nash equilibrium in multistage games both in the cooperative case and the zero-sum two-player case. Those two cases used to be treated as different agendas since the seminal paper of Shoham & *al.* [31] and we expect our work to serve as a milestone to reconcile them going further than normal form games [23, 17]. Our first contribution is to propose two novel on-line and decentralized algorithms inspired by actor-critic architectures for Markov Games, each of them working on two timescales. Those two algorithms perform the same actors' update but use different methods for the critic. The first one performs an off-policy control step whilst the second relies on a policy evaluation step. Although the actor-critic architecture is popular for its success in solving (continuous action) RL domains, we choose this architecture for a different reason. Our framework requires handling non-stationarity (because of adaptation of the other players) which is another nice property of actor-critic architectures. Our algorithms are stochastic approximations of two dynamical systems that generalize the work of [23] and [17] on the fictitious play process from normal form games to multistage games [7].

In the following, we first outline related work (in Sec. 2) and then describe the necessary background in both game theory and RL (Sec. 3) to introduce our first contribution, the two-timescale algorithms (Sec. 4). These algorithms are stochastic approximations of two continuous-time processes defined in Sec. 5. Then, we study (in Sec. 5) the asymptotic behavior of these continuous-time processes and show, as a second contribution, that they converge in self-play in cooperative games and in zero-sum two-player games. In Sec. 6, our third contribution proves that the algorithms are stochastic approximations of the two continuous-time processes. Finally, we perform an empirical evaluation (in Sec. 7).

2 Related Work

Decentralized reinforcement learning in games has been studied widely in the case of normal form games and includes regret minimization approaches [9, 12] or stochastic approximation algorithms [23]. However, to our knowledge, none of the previous methods have been extended to independent reinforcement learning in Markov Games or any intermediate models such as MSGs with guarantees of convergence both for cooperative and zero-sum case. Finding a single independent RL algorithm addressing both cases is still treated as separate agendas since the seminal paper [31].

Q-Learning Like Algorithms: The adaptation of RL algorithms to the multi-agent setting was the first approach to address online learning in games. On-line algorithms like Q-learning [32] are often used in cooperative multi-agent learning environments but fail to learn a stationary strategy in simultaneous zero-sum two-player games. They fail in this setting because, in simultaneous zero-sum two-player games, it is not sufficient to use a greedy strategy to learn a Nash equilibrium. In [25], the Q-learning method is adapted to guarantee convergence to zero-sum two-player MGs. This method isn't an independent learning algorithm anymore as each player needs to observe the action of the opponent. Other adaptation to N-player games were developed [18, 14]. However, all these methods require the observation of the opponents' action and the two last ones are guaranteed to converge only under very conservative hypotheses. Moreover, these are not decentralized methods as each player needs to observe the reward of the others.

Independent Policy Gradient: Independent policy gradient methods is also an other attempt to address the problem of learning in games. In MGs and in normal form games, the policy hill climbing method [8] is probably the first approach with theoretical guarantees. But the guarantees of convergence are often limited to the two player two actions setting and fail to converge in zero-sum games. In this approach, each agent follows a gradient ascent on its expected outcome. The behavior of this algorithm can be improved using heuristics on the learning rate as reported in [8]. Many attempt were made to build on this approach such as the GIGA-WoLF algorithm but again, convergence properties are limited to the two player two actions case. It can also scale up using function approximation as in [2] which results in an actor-critic like method and fits in our setting.

MCTS Algorithms in MSGs and Conterfactual Regret Minimization Algorithms (CFR): A trove variety of tree search algorithms exists to compute Nash equilibria in MSGs. A review of those algorithms can be found in [7] and mainly focuses in zero-sum two-player simultaneous move multistage games. The convergence guarantees of those methods were first analyzed in [24] and then detailed in [19]. Those algorithms require that one can query the model at each stage of the game and as such do not belong to the family of independent RL methods.

An other family of algorithms related to our work is the CFR type of algorithms and its wide number of variant. Even if these algorithms have sample-based variations [20], they usually require each agent to be aware of the opponent's strategy and some game specific information. Only one variation of MCCFR was suggested in [20] would match our setting but is still unexplored in the literature and its unclear how function approximation could be used in this setting.

Two-timescale algorithms in MDPs and in MGs: Two-timescale algorithms have been the subject of a wide literature in MDPs that starts from the seminal work of [5]. These works mainly analyze the use of linear function approximation. However, when applied independently in a multiagent setting, those algorithms no longer have convergence guarantees (even without function approximation). In [27], the authors attempt to provide an on-line model-free algorithm with guarantees. This approach requires each player to observe the reward of the others. The authors claim that their algorithm is guaranteed to converge to a Nash equilibrium but the proof is broken (details in App.K).

3 Background

Markov Games (MGs): Formally, an MG is a tuple $< N, S, \boldsymbol{A}, p(s'|s,\boldsymbol{a}), \boldsymbol{r}(s,\boldsymbol{a}) >$. In each state $s \in S$ of the MG, each of the N-players simultaneously takes an action $a^i \in A^i$. The joint action $\boldsymbol{a}$ is $(a^1, \dots, a^N)$ or $(a^i, \boldsymbol{a}^{\boldsymbol{\text{-}i}})$ (where $\boldsymbol{a}^{\boldsymbol{\text{-}i}}$ is the tuple $(a^1, \dots, a^{i-1}, a^{i+1}, \dots, a^N)$) and the joint action set $\boldsymbol{A}$ is $A^1 \times \cdots \times A^N = A^i \times \boldsymbol{A}^{\boldsymbol{\text{-}i}}$. As a result of this joint action $\boldsymbol{a}$, player i collects a reward $r^i(s,\boldsymbol{a})$ $(\boldsymbol{r}(s,\boldsymbol{a}) = (r^1(s,\boldsymbol{a}), \dots, r^N(s,\boldsymbol{a}))$ and move to a next state s' following the probability $p(s'|s,\boldsymbol{a})$. In such a setting, the usual goal is to find a strategy for each player that maximizes some long term reward. A strategy for player i is a function from state s to a distribution over actions A^i and is written $\pi^i(.|s)$. The joint strategy will be written $\boldsymbol{\pi} = (\pi^1, \dots, \pi^N) = (\pi^i, \boldsymbol{\pi}^{\boldsymbol{\text{-}i}})$. When the strategy of each player is fixed, the MG behaves like a Markov chain of kernel $\mathcal{P}_{\boldsymbol{\pi}}(s'|s) = E_{\boldsymbol{a}\sim\boldsymbol{\pi}}[p(s'|s,\boldsymbol{a})]$ (note that $\mathcal{P}_{\boldsymbol{\pi}}$ can be seen as a squared matrix of size $|S| \times |S|$). The reward associated with that Markov chain is averaged over the joint strategy $r^i_{\boldsymbol{\pi}}(s) = E_{\boldsymbol{a}\sim\boldsymbol{\pi}}[r^i(s,\boldsymbol{a})]$. When the strategy of the i^{th} player's opponents is fixed, the process is reduced to an MDP of kernel $p_{\boldsymbol{\pi}^{\text{-}i}}(s'|s,a^i) = E_{\boldsymbol{a}^{\text{-}i}\sim\boldsymbol{\pi}^{\text{-}i}}[p(s'|s,a^i,\boldsymbol{a}^{\boldsymbol{\text{-}i}})]$ and reward function $r^i_{\boldsymbol{\pi}^{\text{-}i}}(s,a^i) = E_{\boldsymbol{a}^{\text{-}i}\sim\boldsymbol{\pi}^{\text{-}i}}[r^i(s,a^i,\boldsymbol{a}^{\boldsymbol{\text{-}i}})]$.

Multistage Game: We consider games that can be modeled as trees (see Figure 1). In these games, the interaction between players start in an initial state $\tilde{s}$ and proceeds in stages as illustrated in Figure 1 and terminates at state Ω (a formal definition is given in Section A of the appendix).

Value Function: In a γ-discounted multistage game, (with $\gamma \in (0,1)$), each player's goal is to maximize the rewards it accumulates starting from any state s. This value is defined as follows:

$$v^i_{\boldsymbol{\pi}}(s) = E[\sum_{t=0}^{+\infty} \gamma^t r^i_{\boldsymbol{\pi}}(s_t)|s_0 = s, s_{t+1} \sim \mathcal{P}_{\boldsymbol{\pi}}(.|s_t)], \qquad (1)$$

and,

$$v^i_{\boldsymbol{\pi}} = \left(\mathcal{I} + \gamma\mathcal{P}_{\boldsymbol{\pi}} + \cdots + \gamma^{|S|}\mathcal{P}^{|S|}_{\boldsymbol{\pi}}\right) r^i_{\boldsymbol{\pi}}. \qquad (2)$$

Note that for a multistage game, the value function is well defined even for $\gamma = 1$ since the process has an

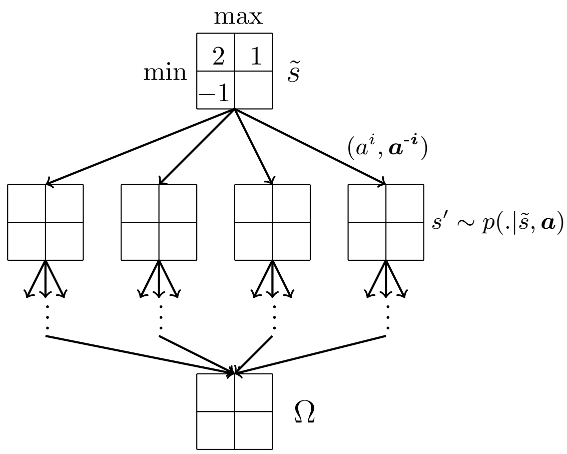
max
min
2
1
−1
$\tilde{s}$
$(a^i, \boldsymbol{a}^{-i})$
$s' \sim p(.|\tilde{s}, \boldsymbol{a})$
Ω

Figure 1: Example of a two-player zero-sum multistage game with deterministic dynamics.

absorbing state Ω where the reward function is null. For $t > |S|$, we have that $\mathcal{P}^t_{\boldsymbol{\pi}} r^i_{\boldsymbol{\pi}} = 0$ since after $|S|$ steps (at most) the process ends up in the absorbing state Ω. The matrix $\mathcal{I}$ is the identity application.

State-Action Value Function: we can also define a state-action value function per player, that is the value of performing action a^i in state s and then following its strategy:

$$Q^i_{\boldsymbol{\pi}}(s, a^i) = r^i_{\boldsymbol{\pi}^{-i}}(s, a^i) + \gamma \sum_{s'} p_{\boldsymbol{\pi}^{-i}}(s'|s, a^i) v^i_{\boldsymbol{\pi}}(s'). \tag{3}$$

Logit Choice Function: when given a choice between actions $a \in A$ based on an expected value $Q(a)$ (in the following $Q(a)$ will be the state-action value function $Q^i_{\boldsymbol{\pi}}(s, a^i)$ or $Q^{*i}_{\sigma,\boldsymbol{\pi}^{-i}}(s, a^i)$ defined below), it is common not to take the action that maximizes $Q(a)$ but to choose suboptimal actions so as to favor exploration. One common choice of suboptimal actions is to pick them according to the logit choice function $B_\eta(Q)(a) = \frac{\exp(\eta^{-1}Q(a))}{\sum_{\tilde{a}} \exp(\eta^{-1}Q(\tilde{a}))}$. We will write $C_\eta(Q) = \sum_{\tilde{a} \in A} Q(\tilde{a}) B_\eta(Q)(\tilde{a})$ the expected outcome if actions are taken according to the logit choice function. This definition can be generalized to the notion of choice probability function (see Section B in the appendix) and will be written $B_\sigma(Q)$ and $C_\sigma(Q)$ where σ is a function in $\mathbb{R}^{\Delta A}$. Here ΔA is the set of distributions over actions. In the case of the logit choice function, η is linked to σ through the following formula $\sigma(\pi) = \sum_a \eta \pi(a) \ln(\pi(a))$.

Best Responses: If the strategy $\boldsymbol{\pi}^{-i}$ of all opponents is fixed, the value of the best response is $v^{*i}_{\sigma,\boldsymbol{\pi}^{-i}}$ and is recursively defined as follows (starting from Ω and going in increasing order with respect to ϕ): $v^{*i}_{\sigma,\boldsymbol{\pi}^{-i}}(s) = C_\sigma(r^i_{\boldsymbol{\pi}^{-i}}(s, a^i) + \sum_{s' \in S} p_{\boldsymbol{\pi}^{-i}}(s'|s, a^i) v^{*i}_{\sigma,\boldsymbol{\pi}^{-i}}(s'))$. From the definition of that value function, we define the corresponding Q-function:

$$Q^{*i}_{\sigma,\boldsymbol{\pi}^{-i}}(s, a^i) = r^i_{\boldsymbol{\pi}^{-i}}(s, a^i) + \gamma \sum_{s'} p_{\boldsymbol{\pi}^{-i}}(s'|s, a^i) v^{*i}_{\sigma,\boldsymbol{\pi}^{-i}}(s'). \tag{4}$$

Bellman Operators: We define the two following Bellman operators on the Q-function:

$$\begin{aligned}\left[T^i_{\boldsymbol{\pi}} Q^i\right](s, a^i) &= r^i_{\boldsymbol{\pi}^{-i}}(s, a^i) \\ &+ \gamma \sum_{s'} p_{\boldsymbol{\pi}^{-i}}(s'|s, a^i) E_{b^i \sim \pi^i(.|s')} \left[Q^i(s', b^i)\right], \end{aligned} \tag{5}$$

and

$$\begin{aligned}[T^{*i}_{\sigma,\boldsymbol{\pi}^{-i}} Q^i](s, a^i) &= r^i_{\boldsymbol{\pi}^{-i}}(s, a^i) \\ &+ \gamma \sum_{s'} p_{\boldsymbol{\pi}^{-i}}(s'|s, a^i) C_\sigma \left(Q^i(s', .)\right). \end{aligned} \tag{6}$$

The value $Q^i_{\boldsymbol{\pi}}$ is the fixed point of the operator $T^i_{\boldsymbol{\pi}}$ and $Q^{*i}_{\sigma,\boldsymbol{\pi}^{-i}}$ is the fixed point of the operator $T^{*i}_{\sigma,\boldsymbol{\pi}^{-i}}$. Furthermore, a strategy $\tilde{\pi}^i$ is greedy with respect to a Q-function Q^i if $T^{*i}_{\sigma,\boldsymbol{\pi}^{-i}} Q^i = T^i_{\tilde{\pi}^i,\boldsymbol{\pi}^{-i}} Q^i$.

Operators on the Value Function: we define the counterparts of those operators on value functions as:

$$[T^i_{\boldsymbol{\pi}} v^i](s) = r^i_{\boldsymbol{\pi}}(s) + \gamma \sum_{s'} p_{\boldsymbol{\pi}}(s'|s) v^i(s') = r^i_{\boldsymbol{\pi}} + \gamma \mathcal{P}_{\boldsymbol{\pi}} v^i \tag{7}$$

and

$$[T^{*i}_{\sigma,\boldsymbol{\pi}^{-i}} v^i] = C_\sigma(r^i_{\boldsymbol{\pi}^{-i}}(s, .) + \gamma \sum_{s'} p_{\boldsymbol{\pi}^{-i}}(s'|s, .) v^i(s')). \tag{8}$$

From these definitions, we have the two following value functions:

$$v^{*i}_{\sigma,\boldsymbol{\pi}^{-i}}(s) = C_\sigma(Q^{*i}_{\sigma,\boldsymbol{\pi}^{-i}}(s, .)) \tag{9}$$

and

$$v^i_{\boldsymbol{\pi}}(s) = E_{b^i \sim \pi^i(.|s)} \left[Q^i_{\boldsymbol{\pi}}(s, b^i)\right]. \tag{10}$$

Smooth Nash Equilibrium: The goal in this setting is to find a strategy π^i for each player that recursively (in increasing order with respect to function $\phi(.)$) fulfills the following condition: $\forall i,\ E_{a^i \sim \pi^i(.|s)} \left[Q^i_{\boldsymbol{\pi}}(s, .)\right] = C_\sigma(Q^i_{\sigma,\boldsymbol{\pi}}(s, .))$. As a consequence, we have that $Q^{*i}_{\sigma,\boldsymbol{\pi}^{-i}} = Q^i_{\boldsymbol{\pi}}$ and $v^i_{\boldsymbol{\pi}} = v^{*i}_{\sigma,\boldsymbol{\pi}^{-i}}$. Furthermore, in the case of a zero-sum two-player game, we have $v^i_{\boldsymbol{\pi}} = -v^{-i}_{\boldsymbol{\pi}}$ (where $v^{-i}_{\boldsymbol{\pi}}$ is the value function of the opponent)

4 Actor-Critic Fictitious Play

We present here our first contribution: the actor-critic fictitious play algorithm for MGs. This is an on-line and decentralized process. At each time-step n, all players are in state s_n, players choose independently an action a^i_n according to their current strategy $\pi^i_n(.|s_n)$ and observe independently one from another a reward signal $r^i_n = r^i(s_n, \boldsymbol{a}_n)$. The process is decentralized, meaning players do not observe others' actions nor their rewards. Then, the game moves to the following state s_{n+1}. If the process reaches the absorbing state Ω, we simply restart from the beginning $(\tilde{s})$.

Algorithm 1 On-line Actor Critic Fictitious Play

```
Input: An initial strategy $\pi^i_0$ and an initial value $v^i_0 = 0$. Two learning rates $\{\alpha_n\}_{n\geq 0}$, $\{\beta_n\}_{n\geq 0}$ satisfying assumption A 3 and an initial state $s_0 = \tilde{s}$.
for n=1,2,... do
    Agent $i$ draws action $a^i_n \sim \pi^i(.|s_n)$.
    Agent $i$ observes reward $r^i_n = r^i(s_n, \boldsymbol{a}_n)$.
    Every player observes the next state $s_{n+1} \sim p(.|s_n, \boldsymbol{a}_n)$.
    actor step
    $\pi^i_{n+1}(s_n, .) = (1 - \beta_n)\pi^i_n(s_n, .) + \beta_n B_\sigma(Q^i_n(s_n, .))$
    critic step
    Either an off-policy control step:
    $Q^i_{n+1}(s_n, a^i_n) = (1 - \alpha_n)Q^i_n(s_n, a^i_n) + \alpha_n \left(r^i_n + C_\sigma\left(Q^i_n(s_{n+1}, .)\right)\right)$
    Or a policy evaluation step:
    $Q^i_{n+1}(s_n, a^i_n) = (1 - \alpha_n)Q^i_n(s_n, a^i_n) + \alpha_n \left(r^i_n + E_{b\sim\pi^i_n(.|s_{n+1})}\left(Q^i_n(s_{n+1}, b)\right)\right)$
    if $s_{n+1} = \Omega$ then
        $s_{n+1} = \tilde{s}$.
    end if
end for
Return The joint strategy $\boldsymbol{\pi}$ and values $v^i$ for all $i$.
```

The learning algorithm performs two updates. First, it updates the players' strategy (actors' update). The strategy π^i_{n+1} is a mixture between the current strategy π^i_n and either a local best response $B_\sigma(Q^i_{\boldsymbol{\pi}_n}(s_n, .))$ or a global best response $B_\sigma(Q^{*i}_{\sigma,\boldsymbol{\pi}^{-i}_n}(s_n, .))$. The actors' update is performed according to a slow timescale β_n. Second, it performs the critics' update which evaluates the current strategy. It happens on a fast timescale α_n on which we can consider that the strategy of every player is stationary. If at the actors' step we want to act according to a local best response dynamics, the critic step will perform a policy evaluation step. If we want to perform a global best response dynamics, the critic will perform an off-policy evaluation step. Thus, at the slow timescale, the Q-function Q^i_n has almost converged to $Q^{*i}_{\sigma,\boldsymbol{\pi}^{-i}_n}$ or to $Q^i_{\boldsymbol{\pi}_n}$. Therefore, we obtain canonically two algorithms.

In the next section, as a second contribution, we introduce the two dynamical processes corresponding to these algorithms and we prove that they possess desirable properties (*i.e.* rationality and convergence in self play for zero-sum two-player and cooperative games). The proofs rely non trivial techniques to propagate the Lyapunov stability property of the Fictitious play process of these ODE on the tree structure of the multistage games. Then in Sec. 6, as a third contribution, we show formally that these two algorithms (Algo.1) are stochastic approximations of those dynamical processes. This is again non-trivial as we need to prove the convergence of a two-time scale discrete scheme depending on a Markov chain.

5 Fictitious play in Markov Games

In this section, we propose novel definitions for two perturbed best response dynamics in the case of MGs. These dynamics are defined as a set of Ordinary Differential Equations (ODE) that generalizes the continuous time Fictitious play process to MSGs [17]. Then, we prove the convergence of these processes in multistage games. To do so, we build on the work of [17] on the stability of the Fictitious play process in normal form games. By induction on the tree structure of multistage games, we prove that our processes have stable attractors in zero-sum two-player and in cooperative multistage games. Later in Sec. 6, we will prove that our actor-critic algorithms track the solutions of that ODE and thus are guaranteed to converge in both settings. The first one considers a local best response dynamics:

$$\dot{\pi}^i_t(.|s) = d_{\boldsymbol{\pi}_t}(s)[B_\sigma(Q^i_{\boldsymbol{\pi}_t}(s, .)) - \pi^i_t(.|s)] \qquad (11)$$

The second process considers a global best response:

$$\dot{\pi}^i_t(.|s) = d_{\boldsymbol{\pi}_t}(s)[B_\sigma(Q^{*i}_{\sigma,\boldsymbol{\pi}^{-i}_t}(s, .)) - \pi^i_t(.|s)] \qquad (12)$$

Two properties are usually desirable for such a stochastic process which are *rationality* and *convergence* in self-play [8]. Rationality implies that if other players converge to a stationary strategy, the learning algorithm will converge to a best response strategy. The convergence property received many definitions in the MARL literature and usually ensures convergence of the algorithm against a class of other algorithms or, as studied here, the convergence in self-play.

Rationality: For the rest of this paragraph, let us study a fixed player i. First, if we consider the case where other players are stationary (*i.e.* $\boldsymbol{\pi}_t^{\text{-}i} = \boldsymbol{\pi}^{\text{-}i}$), the strategy of player i will converge to $B_\sigma(Q^{*i}_{\sigma,\boldsymbol{\pi}^{\text{-}i}}(s,.))$ for the second process Eq. (12) (since $B_\sigma(Q^{*i}_{\sigma,\boldsymbol{\pi}^{\text{-}i}}(s,.))$does not depend on π_t^i the solution of the second process converges exponentially toward the best response strategy). For the first process, let us show that if the strategy π^i follows the dynamics described by (11), it converges to a best response strategy. The proof of this property requires the following two technical lemmas.

(T, δ)-perturbation: Let us consider the following ODE where $h(.) : \mathbb{R}^n \to \mathbb{R}^n$ is a Lipschitz continuous function: $\dot{z}(t) = h(z(t))$ (13)

We consider the case where ODE (13) has an asymptotically stable attractor set J. A bounded measurable function $y(.) : \mathbb{R}^+ \to \mathbb{R}^N$ is a (T, δ)-perturbation of (13) $(T, \delta > 0)$ if there exist $0 = T_0 < T_1 < T_2 < \cdots < T_n$ with $T_{i+1} - T_i \geq T$ and solutions $z^j(t),\ t \in [T_j, T_{j+1}]$ of (13) for all $j \geq 0$ such that $\sup\limits_{t\in[T_j,T_{j+1}]} \|z^j(t) - y(t)\| < \delta$. The following technical lemma can be proved [6].

Lemma 1. *Given $\epsilon, T > 0$, there exists a $\bar{\delta} > 0$ such that for $\delta \in (0, \bar{\delta})$, every (T, δ)-perturbation of Eq.* (13) *converges to J^ϵ (def: the ϵ-neighborhood of J).*

The next lemma is used all over this section. It proves that if, instead of following a given process $\dot{x}(t) = f(x(t))$, one follows a perturbed version of that process $\dot{y}(t) = f_t(y(t))$ one converges to the set of stable equilibria of the unperturbed process if $f_t(.)$ converges toward $f(.)$.

Lemma 2. *Let's study the two following ODE $\dot{x}(t) = f(x(t))$ and $\dot{y}(t) = f_t(y(t))$ and let's assume that $\limsup \|f(.) - f_t(.)\| \to 0$ uniformly. Furthermore, let's assume that f is Lipschitz continuous. Then, any bounded solution of $\dot{y}(t) = f_t(y(t))$ converges to the set of attractors of $\dot{x}(t) = f(x(t))$.*

Proof. The full proof of this lemma is left in appendix C. □

Now, we show that if player i follows the direction given by the local best response dynamics Eq. (11), it converges to a best response to $\boldsymbol{\pi}^{\text{-}i}$.

Proposition 1. *In any game, if the strategy $\boldsymbol{\pi}^{\text{-}i}$ of the opponents is fixed, the process* (11) *converges to a best response.*

Proof. The proof of this proposition is left in appendix G □

Now, let's consider the case where the strategy of other players $\boldsymbol{\pi}_t^{\text{-}i}$ converges to a stationary strategy $\boldsymbol{\pi}^{\text{-}i}$. In that case, one can show that, for a fixed i, the solutions of ODE (11) and (12) are the same as if $\boldsymbol{\pi}_t^{\text{-}i}$ was fixed.

Proposition 2. *If $\boldsymbol{\pi}_t^{\text{-}i} \to \boldsymbol{\pi}^{\text{-}i}$, then the solutions of ODE* (11) *and* (12) *are the same as the one of:*

$$\dot{\pi}_t^i(.|s) = d_{\pi_t^i,\boldsymbol{\pi}^{\text{-}i}}(s)[B_\sigma(Q^i_{\pi_t^i,\boldsymbol{\pi}^{\text{-}i}}(s,.)) - \pi_t^i(.|s)]$$

and

$$\dot{\pi}_t^i(.|s) = d_{\pi_t^i,\boldsymbol{\pi}^{\text{-}i}}(s)[B_\sigma(Q^{*i}_{\sigma,\boldsymbol{\pi}^{\text{-}i}}(s,.)) - \pi_t^i(.|s)]$$

Proof. The proof of this proposition is left in appendix H □

Thus, if all other players converge to a stationary strategy, the strategy of player i converges to a best response to $\boldsymbol{\pi}^{\text{-}i}$.

Convergence in Multistage Games: Concerning the convergence in self-play of those two fictitious play processes (ODE (11) and (12)), one must first be aware that they will not converge in all multistage games. Even in normal form games, the fictitious play process is known to be unstable in some cases. For example, the fictitious play process is known to oscillate in some two-player games [30]. More surprisingly, [15] proved in the case of normal form games that there exists no uncoupled dynamics which guarantees Nash convergence. In their paper, they consider dynamics for each player which only depends on its own reward and possibly on the strategy of all players. They present an example in which this class of dynamical systems has no stable equilibrium. We were able to prove the convergence of our process in the case of zero-sum two-player and cooperative multistage games.

Proposition 3. *In a zero-sum two player multistage game, the process* (11) *and the process* (12) *converge to a smooth Nash equilibrium $\boldsymbol{\pi}$.*

Proof. The proof of this proposition is left in appendix I □

Remark 1. *Proposition 3 can be adapted to study the cooperative case (i.e. when players receive the same reward $\forall i, j\ r^i(s, \boldsymbol{a}) = r^j(s, \boldsymbol{a})$). The proof in the cooperative case also works by induction. The corresponding proposition and proof can be found in the appendix E.*

6 Stochastic Approximation with Two-Timescale

In this section, we provide a novel stochastic approximation theorem taking advantage of two independent techniques. First, we use two timescales because the process we are studying (defined on the strategy of each player called the *actor*) has a complex dependency on the strategy of all players through the Q-function (the *critic*). In an on-line setting, one can't have access to the Q-function of a given strategy (either $Q^i_{\boldsymbol{\pi}}$ or $Q^{*i}_{\sigma,\boldsymbol{\pi}^{\text{-}i}}$) as it is an asymptotic solution of the critic process. Thus formally, we look for an asymptotically stable solution of $\dot{y}(t) = g(\mu(y(t)), y(t))$ (actors' update) where $\mu(y)$ is the stable solution of another process $\dot{x}(t) = f(x(t), y)$(critics' update). In our case $\mu(y)$ is either $Q^i_{\boldsymbol{\pi}}$ or $Q^{*i}_{\sigma,\boldsymbol{\pi}^{\text{-}i}}$, y is the strategy and x the action-value function. Instead of waiting until the subroutine converges (the critic part) to iterate over the main routine (the actor part), [6] gives an elegant solution to that class of problems by using two timescales to update simultaneously x_n and y_n. The process x_n will be updated according to a "fast" timescale α_n. On that timescale, y_n behaves as if it was stationary and thus, x_n is an estimate of $\mu(y_n)$ (i.e. $\|\mu(y_n) - x_n\| \to 0$). The process y_n will move on a "slower" timescale β_n. On that slower timescale, one can treat the process x_n as $\mu(y_n)$ and thus, y_n will converge to a stable solution of $\dot{y}(t) = g(\mu(y(t)), y(t))$.

Second we use an averaging technique. Our process is on-line and follows a Markovian dynamics of stationary distribution $d_{\boldsymbol{\pi}_t}$ controlled by the policy $\boldsymbol{\pi}_n$. All policies and Q-functions are only updated on state s_n. Thus, the two processes x_n and y_n also depend on a Markov process Z_n controlled by y_n. Again, [3] shows that, in the case of a simple timescale, the process $y_{n+1} = y_n + \alpha_n f(y_n, Z_n)$ tracks the solution of $\dot{y}(t) = \bar{f}(y(t), d_{y(t)})$ where $\bar{f}(y(t), d_{y(t)})$ is the average of f over the stationary distribution of Z_n (the distribution d_y). Formally, $\bar{f}(y, d_y) = \sum_{z\in\mathcal{Z}} f(y, z)d_y(z)$. In a nutshell, we generalize that result on stochastic approximation with a controlled Markov noise to two timescale stochastic approximation. In our case, we need to study the following recursion:

$$x_{n+1} = x_n + \alpha_n f(x_n, y_n, Z_n), \tag{14}$$

$$y_{n+1} = y_n + \beta_n g(x_n, y_n, Z_n), \tag{15}$$

Where:

A 1. *Functions f and g are jointly continuous in their arguments and Lipschitz in their two first arguments uniformly with respect to the third.*

A 2. *The controlled Markov process Z_n takes its value in a discrete space $\mathcal{Z}$ controlled by variable y_n. The variable Z_{n+1} follows the kernel $p(.|Z_n, y_n)$ which is uniformly continuous in y_n. Furthermore, let us suppose that if $y_n = y$, the Markov chain Z_n has a unique invariant distribution $d_y(.)$.*

We note $\bar{f}(x, y, d_y) = \sum_{z\in\mathcal{Z}} f(x, y, z)d_y(z)$ the function f averaged over the stationary distribution of the Markov chain defined in the previous assumption. Similarly, $\bar{g}(x, y, d_y) = \sum_{z\in\mathcal{Z}} g(x, y, z)d_y(z)$.

A 3. *The sequences $\{\alpha_n\}_{n\geq 0}$ and $\{\beta_n\}_{n\geq 0}$ are two positives decreasing step-size sequences satisfying: $\sum\limits_{n\geq 0} \alpha_n = \sum\limits_{n\geq 0} \beta_n = \infty$, $\sum\limits_{n\geq 0} \alpha_n^2 < \infty$, $\sum\limits_{n\geq 0} \beta_n^2 < \infty$ and $\beta_n = o(\alpha_n)$*

A 4. *We need $\sup\limits_n \|x_n\| < \infty$ and $\sup\limits_n \|y_n\| < \infty$*

A 5. *For any constant y, the ODE: $\dot{x}(t) = \bar{f}(x(t), y, d_y)$, has a globally asymptotically stable equilibrium $\mu(y)$. That stable equilibrium $\mu(.)$ is a Lipschitz continuous function.*

A 6. *The ODE $\dot{y}(t) = \bar{g}\left(\mu\left(y(t)\right), y(t), d_{y(t)}\right)$ has a globally asymptotically stable equilibrium y^**

Theorem 1. *$(x_n, y_n) \to (\mu(y^*), y^*)$ almost surely.*

Analysis of Actor-Critic Fictitious Play: The analysis of our two algorithms is a direct application of the previous theorem and is left in appendix (App. J). In a nutshell, the critic Q_n is updated on a fast timescale (i.e. x_n) and we will prove that it's recursion satisfy assumption A 1, A 2, A 3, A 4, and A 5. The actor π_n is updated on a slow timescale (i.e. y_n) and satisfy assumption A 1, A 2, A 3, A 4, and A 6. Thanks to the use of choice probability function, assumption A 1 is satisfied on the slow timescale since a choice probability function is lipschitz with respect to the Q-function.

7 Experiment on Alesia

Alesia [26], which resembles the Oshi-Zumo game in [10, 7] (meaning "the pushing sumo"), is a two-player board game where each player starts with N coins. They are positioned in the middle of a board with $2K+1$ different positions (see Figure 2). At each round, each player chooses secretly a certain amount of coins to bid (let's say (a^0, a^1)). If the player's budget is not null, s/he has to bet at least one coin. If player 0's bid is larger (respectively smaller) than the one of player 1, player 0 (respectively player 1) moves toward his side of the board. If the bids are equal, then players will stay on their current position. This process continues until players have no coins left or until one player reaches one side of the board. The final position determines the winner. The game ends with a draw if no one succeeded to reach his side of the board. If player 1 (respectively player zero) reaches his side of the board, the reward is $(-1, 1)$ (respectively $(1, -1)$).

This game is challenging in many aspects. First, rewards are sparse and are received at the end of the game. Furthermore, strategies need to be stochastic [10, 7]. The performance criterion used to compare algorithms is the difference of the value of the

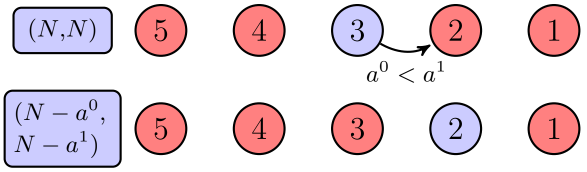

Figure 2: Alesia rules for $K = 2$.

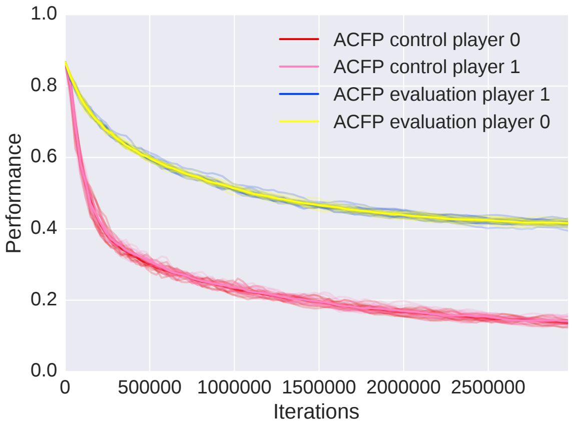

Figure 3: Performance of the strategy π^i (y-axis) along iterations (x-axis).

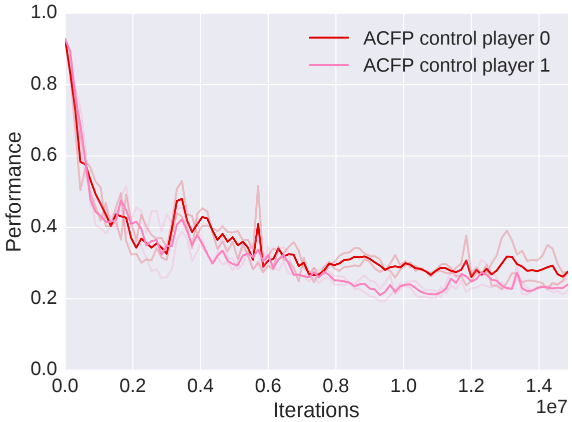

Figure 4: Performance of the strategy π^i (y-axis) along iterations (x-axis).

joint strategy $v^i_{\boldsymbol{\pi}}(\tilde{s})$ and the value of the best response $v^i_{0,\boldsymbol{\pi}^{-i}}(\tilde{s})$ without any perturbation σ (*i.e.* $\sigma = 0$)

We ran experiments for a game with learning rates of the form $\alpha_n = \frac{\alpha_0 a_\alpha}{a_\alpha + n^{b_\alpha}}$ and $\beta_n = \frac{\beta_0 a_\beta}{a_\beta + n^{b_\beta}}$. We used a logit choice function with η decaying with the number of iterations $\eta_n = \frac{\eta_0 a_\eta}{a_\eta + n}$. The two step-sizes α_n and β_n were chosen to satisfy the conditions of Theorem 1. The experiments were ran with an initial budget of $N = 10$ and a board of size $K = 3$. The step-size parameters were $(\alpha_0, a_\alpha, b_\alpha) = (0.1, 10^4, 0.8)$, $(\beta_0, a_\beta, b_\beta) = (0.01, 10^4, 1.0)$ and $(\eta_0, a_\eta) = (0.1, 10^6)$. The results are displayed in figure 3 and shows that both algorithms learn a strategy that tend to be closer to its own best response over learning (meaning that both algorithms learn a Nash equilibrium). An empirical finding is that ACFP with a control critic converges faster than the one with a policy evaluation critic on that game.

Function approximation: We also ran these experiments with function approximations both on the strategy and on the Q-function. The main issue in Alesia is that the number of actions available to each player depends on the remaining budget of the player. For each player, we aggregated states for which the budget of the player is fixed. The aggregated states are consecutive states with respect to the budget of the other players'. The results reported in Figure 4 show that our method is robust to this form of function approximation. We ran experiment with an initial budget of $N = 15$ on which the algorithm was able to learn a minimax strategy. The step-size parameters were $(\alpha_0, a_\alpha, b_\alpha) = (0.1, 10^5, 0.9)$, $(\beta_0, a_\beta, b_\beta) = (0.01, 10^5, 1.0)$ and $(\eta_0, a_\eta) = (0.05, 10^8)$.

8 Conclusion

To summarize, we defined two novel algorithms. These algorithms are on-line and decentralized procedures that provably converge in two classes of multistage games: zero-sum two-player and cooperative games. Compared to previous family of methods [8, 25], these algorithms can be applied to a larger number of settings without heuristics. This paper opens several interesting research avenues. The first is the study of the stability of the two dynamical processes in MGs. The proof given for multistage games propagates the convergence property from the end of the game to the initial state. Since MGs have a graph structure, this technique can't be applied. Thus, other stability arguments must be found to ensure convergence. The main contributions of this paper were algorithmic and theoretical but we also proposed an empirical evaluation that provides good evidence that our methods are applicable to real problems and could scale up thanks to function approximation. Whilst there is a wide literature dealing with the off-policy evaluation part or the off-policy control [32], the update of the strategy using general function approximation remains challenging. The main issue is that the strategy space should support convex combination with a best response. Boosting methods such as in [29] could offer a solution. The convex combination of the strategy and the best response could also be done thanks to a classification neural network.

Acknowledgments

We wish to thank Marc Lanctot for the helpful conversations, his comments definitely helped us improving the final version of the paper. This work has received partial funding from CPER Nord-Pas de Calais/FEDER DATA Advanced data science and technologies 2015-2020 and the European Commission H2020 framework program under the research Grant number 687831 (BabyRobot).

References

[1] N. Akchurina. Multiagent Reinforcement Learning: Algorithm Converging to Nash Equilibrium in General-Sum Discounted Stochastic Games. In *Proc. of AAMAS*, 2009.

[2] B. Banerjee and J. Peng. Adaptive policy gradient in multiagent learning. In *Proc. AAMAS*. ACM, 2003.

[3] Vivek S Borkar. Stochastic approximation with controlled markovnoise. *Systems & control letters*, 2006.

[4] Vivek S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2009.

[5] V.S. Borkar. Stochastic Approximation with Two Time Scales. *Systems & Control Letters*, 29(5):291–294, 1997.

[6] V.S. Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 1997.

[7] B. Bošanský, V. Lisý, M. Lanctot, J. Čermák, and M.H.M. Winands. Algorithms for computing strategies in two-player simultaneous move games. *Artificial Intelligence*, 237:1 – 40, 2016.

[8] M. Bowling and M. Veloso. Rational and Convergent Learning in Stochastic Games. In *Proc. of IJCAI*, 2001.

[9] Sbastien Bubeck and Nicol Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[10] M. Buro. *Solving the Oshi-Zumo Game*, pages 361–366. Springer US, 2004.

[11] L. Busoniu, R. Babuska, and B. De Schutter. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2008.

[12] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.

[13] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Proc. of NIPS*, 2016.

[14] A. Greenwald, K. Hall, and R. Serrano. Correlated Q-learning. In *Proc. of ICML*, 2003.

[15] S. Hart and A. Mas-Colell. Uncoupled dynamics do not lead to nash equilibrium. *The American Economic Review*, 2003.

[16] J. Heinrich, M. Lanctot, and D. Silver. Fictitious self-play in extensive-form games. In *Proc. of ICML*, 2015.

[17] J. Hofbauer and W.H. Sandholm. On the global convergence of stochastic fictitious play. *Econometrica*, 70(6):2265–2294, 2002.

[18] J. Hu and M. P. Wellman. Nash Q-Learning for General-Sum Stochastic Games. *Journal of Machine Learning Research*, 4:1039–1069, 2003.

[19] Vojtěch Kovařík and Viliam Lisỳ. Analysis of hannan consistent selection for monte carlo tree search in simultaneous move games. *arXiv preprint arXiv:1509.00149*, 2015.

[20] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte carlo sampling for regret minimization in extensive games. In *Proc. of NIPS*, 2009.

[21] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, karl Tuyls, Julien Perolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Proc. of NIPS*, 2017.

[22] Guillaume J. Laurent, Laëtitia Matignon, and N. Le Fort-Piat. The world of independent learners is not Markovian. *Int. J. Know.-Based Intell. Eng. Syst.*, pages 55–64, 2011.

[23] D.S. Leslie and E.J. Collins. Generalised weakened fictitious play. *Games and Economic Behavior*, 56(2):285–298, 2006.

[24] Viliam Lisy, Vojta Kovarik, Marc Lanctot, and Branislav Bosansky. Convergence of monte carlo tree search in simultaneous move games. In *Proc. of NIPS*, 2013.

[25] M. L. Littman. Markov Games as a Framework for Multi-Agent Reinforcement Learning. In *Proc. of ICML*, 1994.

[26] J. Perolat, B. Scherrer, B. Piot, and O. Pietquin. Approximate Dynamic Programming for Two-Player Zero-Sum Markov Games. In *Proc. of ICML*, 2015.

[27] HL Prasad, Prashanth LA, and Shalabh Bhatnagar. Two-Timescale Algorithms for Learning Nash Equilibria in General-Sum Stochastic Games. In *Proc. of AAMAS*, 2015.

[28] J. Robinson. An iterative method of solving a game. *Annals of mathematics*, pages 296–301, 1951.

[29] B. Scherrer and M. Geist. Local Policy Search in a Convex Space and Conservative Policy Iteration as Boosted Policy Search. In *Proc. of ECML*, 2014.

[30] L.S. Shapley. Some topics in two-person games. *Advances in game theory*, 1964.

[31] Y. Shoham, R. Powers, and T. Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171(7):365 – 377, 2007.

[32] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992.

[33] J. W. Weibull. *Evolutionary game theory.* MIT press, 1997.

[34] Martin Zinkevich, Michael Johanson, Michael H Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Proc. of NIPS*, 2007.