Wind farm control with cooperative multi-agent reinforcement learning

Claire Bizon Monroc[1,2] **Ana Bušić**[1] **Donatien Dubuc**[2] **Jiamin Zhu**[2]
claire.bizon-monroc@inria.fr

[1] Inria and DI ENS
École Normale Supérieure
PSL Research University

[2] IFP Energies nouvelles
Reuil-Malmaison, France
Solaize, France

Abstract

Maximizing the energy production in wind farms requires mitigating wake effects, a phenomenon by which wind turbines create sub-optimal wind conditions for the turbines located downstream. Finding optimal control strategies is however challenging, as high-fidelity models predicting complex aerodynamics are not tractable for optimization. Good experimental results have been obtained by framing wind farm control as a cooperative multi-agent reinforcement learning problem. In particular, several experiments have used an independent learning approach, leading to a significant increase of power output in simulated farms. Despite empirical success, the independent learning approach has no convergence guarantee due to non-stationarity. We show that the wind farm control problem can be framed as an instance of a transition-independent Decentralized Partially Observable Decentralized Markov Decision Process (Dec-POMDP) where the interdependence of agents dynamics can be represented by a directed acyclic graph (DAG). We show that for these problems, non-stationarity can be mitigated by a multi-scale approach, and show that a multi-scale Q-learning algorithm (MQL) where agents update local Q-learning iterates at different timescales guarantees convergence.

1 Introduction

Recent advances in reinforcement learning (RL) have seen a growing interest in solving cooperative multi-agent problems, where several agents interact with the same environment to optimize a common objective [31, 20]. Multi-agent reinforcement learning (MARL) has encountered successes in fields as varied as games with multiple players [2], vehicle routing problem for traffic regulation [32], or recently, distributed optimal control of wind farms [24].

Operating wind turbines causes wind perturbations called wake effects: downstream of the rotor, the velocity of the wind decreases and its turbulence increases. In wind farms, where many wind turbines are grouped together on the same field, wake effects create sub-optimal wind conditions the farm that can reduce the production of downstream turbines. An example of this phenomenon can be seen on Section 1. One solution is to increase the angle between a turbine's rotor and the direction of the wind, called the yaw: this decreases the turbine's own production, but can increase the total power output of the farm by deflecting the wake away from downstream turbines. Finding the optimal yaws to maximize the production is hard. Models must predict complex aerodynamics interactions under turbulent wind inflow and uncertain atmospheric conditions, and the optimal yaws returned by classical optimization approaches based on static models are sensitive to modeling errors.

Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2024).

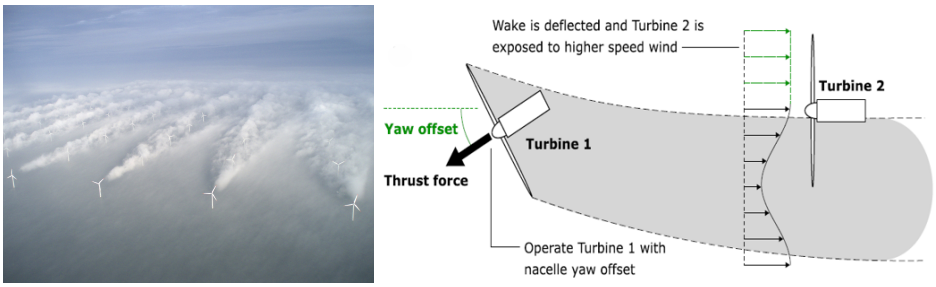


Figure 1: Left: Wake effects in the offshore wind farm of Horns Rev 1 - Vattenfall. Right: Wake steering schema according to [14]. The yaw of the first turbine is increased to deflect the wake away from the downstream turbine.

Instead, recent methods have proposed to frame the wind farm control problem as a cooperative multi-agent reinforcement learning task: in this approach, every turbine is an agent that can choose to increase or decrease its yaw, and all seek to maximize the total power production. Good empirical results have been achieved under this framework, with RL algorithms significantly increasing the total power output of several simulated wind farms [6, 13, 24]. In particular, independent learning, an approach where all agents simultaneously run single-agent RL algorithms, was sufficient to learn optimal memory-less control strategies [11, 30, 8, 6]. To our knowledge no attempt at mobilizing theory to understand this success or derive convergence guarantees has however been made.

In this article, we rely on stochastic approximation techniques to prove the convergence of a multi time-scale approach to tackle the non-stationarity problem. In particular, multi-timescale stochastic approximation showed that several interdependent stochastic processes can all converge when they are updated at different scales [7, 17]. These have been successfully used to build reinforcement learning algorithms maintaining different iterates, to decouple the learning of future rewards and of the best response in various fictitious-play [3, 23, 16] and Q-learning [22]. Unfortunately, these convergence results require the reward function to be stationary, meaning that for a given state-action pair, the collected reward is always sampled from the same distribution. These results do hence not apply to the partial observability case. In [19], a multi-scale approach similar to the one investigated in this article is further evaluated on several multi-agent reinforcement learning problems, and found to stabilize learning in several MARL environments, but an analysis of its convergence is not provided.

Contributions of the paper

- We show that this wind farm control problem **can be framed as a Transition-Independent Dec-POMDP** where agent dynamics are represented as a directed acyclic graph (DAG). This approach identifies a problem structure that can be useful beyond its application to wind farm control.
- We show that for independent learners in a transition-independent Dec-POMDP, **the loss of information due to partial observability can be seen as a Markovian noise**.
- Using weak convergence results for multi-timescale asynchronous updates from [15], we prove that letting agents learn at **different time scales** can be sufficient to **guarantee the convergence** of independent Q-learning **when agent dynamics can be described by a DAG**.
- We build on the network distributed POMDP problems [18], in which interactions between agents can be represented by a sparse graph, and show that our multi-scale Q-learning approach can **exploit known interaction structure to guide learning rates selection**.

The paper is organized as follows. In Section 2 we formalize the problem of finding an equilibrium in a transition-independent Dec-POMDP and show how it can describe our wind farm control problem. Then, in Section 3, we propose a multi-scale Q-learning algorithm and prove its convergence: we lay out the assumption of acyclic dependence structure between agent dynamics, and show how it allows us to apply our multiscale results to the defined class of Dec-POMDP. We then exploit the graph of interaction between agents in a networked problem to derive faster algorithms. Finally, our

experiment in Section 4 evaluates the multi-scale approach on the real industrial problem of wind farm control, and empirically validates its convergence.

2 Preliminaries: cooperative MARL with local learners

Let us consider the case of a fully cooperative, infinite horizon multi-agent reinforcement problem, where state information is distributed among all agents and they must collaborate to maximize a shared reward. Such problems are commonly formulated as decentralized partially observable Markov decision processes (Dec-POMDPs) in the MARL literature.

We further focus on the special case of the *transition-independent* Dec-POMDP [4, 5]. In a transition-independent Dec-POMDP, each agent's local observations only depend on its local actions, so that agents only interact through the shared reward. In general, any blind cooperation problem in which agents must learn to coordinate while being oblivious to each other's existence will fit this description. In the rover exploration problem introduced by [4] for example, several rovers must coordinate to explore a planet. Rovers are assigned distinct sides of the planet to explore so that they do not directly interact, but the value of the information they can gather depends on what is collected by other agents.

We now show that a transition-independent Dec-POMDPs can also be constructed from certain standard Dec-POMDPs, and such a reformulation can be useful to solve real industrial problems. This is the case of the distributed wind farm control problem, in which the local information available to any agent in the Dec-POMDP can be factorized into two components: we have for any agent i, (y^i, w^i) where the first component y^i is the state of the current yaw of the agent, and w^i is a statistic of the local wind conditions at the wind turbine. The first is a private component, that is a local component independent of other agents: the state of each agent's actuator is only controlled and measured by itself. The second is a deterministic function of the private components of other agents, and of an observable exogenous Markovian process that is independent of any agent's action: the local wind conditions are a function of the incoming wind inflow at the entrance of the farm, and of the yaws of other wind turbines. Note that this function is unknown: it is the function predicting the values of the velocity field in front of every turbine rotor, determined by the solutions to 3D Navier-Stokes equations [1]. The identification of such an exogenous process is however sufficient: constructing local states by replacing the second component with a direct observation of the exogenous process frees the local state from dependence on other agents' action, while maintaining the Markovian property of the global MDP, and concludes the reformulation of the problem as a transition-independent Dec-PODMP. In the case of wind farm control, we replace the observations of local wind conditions at every turbine by the observation of a single measure of wind conditions at the entrance of the wind farm w^∞, so that we have in the local observation of each agent (y^i, w^∞). Intuitively, we sacrifice state observability for transition independence.

Let us now formalize the problem of transition-independent Dec-POMDP. We then explicit the assumptions on the transitions and local policies that we will consider in the rest of this paper, before introducing our multi-scale Q-learning algorithm.

2.1 Independent transition Dec-POMDP

We will consider a decentralized partially observable Markov Decision Process (Dec-POMDP) reinforcement learning problem, where M agents interact with the same environment to maximize a common reward. Let us assume a finite state space S and a finite action space A. The global state space S is factorized into M observation or local state spaces $S = S_1 \times \cdots \times S_M$ and for any $s \in S$ we write s^i the corresponding local state in S_i. Note that this means that the local state at any time is a deterministic function of the global state. Similarly, we write $A = A_1 \times \cdots \times A_M$ the factorization of the global action space A. A global reward $r : S \times A \to \mathbb{R}$ is shared by all agents. The reward is bounded in $\mathbb{R}$ by a constant $R > 0$, that is: $\forall (s, a) \in S \times A, |r(s, a)| \leq R$. We write $P : S \times A \times S \to (0, 1)$ a transition kernel, denoting transition probabilities between states given chosen actions. For any state space S and any action space A, we write $\Delta(S, A)$ the set of policies mapping any state $s \in S$ to a distribution over actions in A. Every agent i has a set of local policies $\Delta(S_i, A_i)$, and for any $\pi^i \in \Delta(S_i, A_i)$ we write the probability of taking action a^i in s^i $\pi^i(a^i \mid s^i)$. If the policy is deterministic, so that for any state $s^i \in S_i$ a unique action a^i is chosen with probability one, we directly write $\pi^i(s^i) = a^i$. A global policy π can always be extracted from a set of local

policies $\{\pi^1, \dots, \pi^M\}$ and we write $\pi = (\pi^1, \dots, \pi^M)$. Among all global policies, we thus consider the subset of policies that can be written as a product of local policies $\Pi^o = \times_{i=1}^M \Delta(S_i, A_i)$.

Because any local policy only depends on its local states, we have $\pi(a \mid s) = \Pi_i^M \pi^i(a^i \mid s^i)$ for all a, s. For any discount factor $\beta \in (0,1)$, we consider the maximization of the expectation of the sum of discounted reward, or return $\mathbb{E}_\pi\left[\sum_{k=0}^\infty \beta^k r(s_k, a_k)\right]$, with π a global policy mapping global states to global actions. It has been shown by [10] and [9] that in transition-independent DecPOMDPs, this quantity can be maximized by a the product of local policies in Π^o.

As we consider transition-independent Dec-POMPD, we make the following assumption [4]: every agent's local state is only influenced by its own current state and action.

Assumption 2.1. *We assume that transitions between locally observed states only depend on local state and actions. That is, there are local transition kernels $P^i_{i=1\dots M}$ such that $\forall s, a, s' \in S \times A \times S$*

$$P(s, a, s') = \Pi_{i=1}^M P^i(s^i, a^i, s'^i)$$

For simplicity of notations, we will in the following ignore local states with exogenous processes, but the analysis is easily extended to them.

For any stationary global policy π, the global state process $\mathbf{s}$ is in fact a Markov chain with transition matrix

$$P_\pi(s, s') = \sum_a \pi(a|s) P(s, a, s') = \sum_{a=(a^1,\dots,a^M)} \Pi_{i=1}^M \pi^i(a^i|s^i) P(s, a, s')$$

We now introduce an assumption on the transition function of the MDP.

Assumption 2.2. *For any non-deterministic local policy π^i such that $\forall a^i, s^i \in \mathcal{A}, \pi^i(a^i \mid s^i) > 0$, the local state process is an irreducible and aperiodic Markov chain.*

This classical assumption for Q-learning [28, 27, 12] will ensure that all local state processes admit an invariant distribution, and will converge to it under a fixed policy regardless of the initial distribution. Note that this implies that the global state process is also irreducible and aperiodic.

Using vector notation, we define $d^\pi \in (0,1)^{|S|}$ the invariant distribution over the global state space S satisfying $d^\pi P_\pi = d^\pi$. Similarly, for every agent i we define $d_i^{\pi^i}$, the invariant distribution of the local state process $\mathbf{s}^i$. If we ensure that local policies π^i have non-null probabilities on all the local action space, then **A** 2.2 ensures that the local state-action process $(\mathbf{s}^i, \mathbf{a}^i)$ is also irreducible: it is a Markov chain over $S_i \times A_i$, with transition matrix $P_{\pi^i}((s^i, a^i), (s'^i, a'^i)) = P(s^i, a^i, s'^i)\pi^i(a'^i|s'^i)$ given by **A** 2.1. We denote its invariant distribution as λ^i.

In the rest of this paper, we consider the transition-independent Dec-POMDP that satisfies **A** 2.1 and **A** 2.2.

2.2 Q-functions in a TI Dec-POMDP

For an agent i and a global policy π, we note π^{-i} the set of local policies in π except π^i. For any pair $(s^i, a^i) \in S_i \times A_i$ and any global policy $\pi \in \Pi^o$, we define the i^{th} q-function $Q_{\pi^i}^{\pi^{-i}}(s^i, a^i)$ the value of taking action a^i in local state s^i, and then following policy π^i, provided that any other agent j follows its respective local policy π^j:

$$Q_{\pi^i}^{\pi^{-i}}(s^i, a^i) = \mathbb{E}_{s_0 \sim d^\pi, a_k \sim (\pi^i, \pi^{-i}), s_k \sim P}\left[\sum_{k=0}^\infty \beta^k r(s_k, a_k) \mid s_0^i = s^i, a_0^i = a^i\right] \tag{1}$$

where the initial state s_0 is sampled according to the stationary distribution d^π. These local q-functions $Q_{\pi^i}^{\pi^{-i}}$ can be written as tables of dimension $|S_i| \times |A_i|$, and admit a recursive formula given in lemma 1.

Lemma 1. *Any local q-function eq.* (1) *satisfies the following recursive formula:*

$$\begin{aligned} &Q_{\pi^i}^{\pi^{-i}}(s^i, a^i) \\ &= \sum_s d^\pi(s \mid s^i) \sum_{a^{-i}} \pi^{-i}(a^{-i}|s) r(s, a) + \beta \sum_{s'^i} P^i(s^i, a^i, s'^i) \sum_{a'^i} \pi^i(a'^i|s'^i) Q_{\pi^i}^{\pi^{-i}}(s'^i, a'^i) \end{aligned} \tag{2}$$

The proof of lemma 1 is straightforward but tedious, detailed in appendix A.1. Like for the single-agent q-value function [25], the q-value is split in two parts: an immediate reward collected at the current state and a future gain, that is the reward expectation starting from the next state. Note that at every step the expectation of the reward $r(s, a)$ is taken with regard to a distribution over the global state and the global action. Because the q-value is evaluating the response π^i to π^{-i}, the global action must always be taken with respect to π. Then, per definition of the q-value eq. (1), the initial state is sampled from the stationary distribution d^π. It then follows from the definition of the stationary distribution that the distribution of the next global state will still be d^π, and the local q-value taken at the next step is the expectation of the future gain. We now introduce the definition of a best response, as a local policy π^i which maximizes the return when other local policies are fixed.

Definition 1 (Best response)**.** *A local policy π^i_{br} is said to be a best response to a set of local policies π^{-i} if starting from any local state, it always maximizes the return as the other agents follow local policies π^{-i}. That is, for any local policy π^i we have:*

$$Q^{\pi^{-i}}_{\pi^i_{br}}(s^i, a^i) \geq Q^{\pi^{-i}}_{\pi^i}(s^i, a^i) \quad \forall s^i \in S_i, a^i \in A_i$$

Best response policies will therefore maximize the expectation of this optimal q-value at every state s^i. They can be written as the set of policies π^i such that $\pi^i(\cdot \mid s^i) \in \underset{\rho \in \Omega(s^i)}{\arg\max} \left[\rho^T Q^{\pi^{-i}}_{\pi^i}(s^i, \cdot)\right]$, where $\Omega(s^i) \subset [0, 1]^{|A_i|}$ is the simplex of dimension $|A_i|$ representing the set of local strategies mapping a given local state to a distribution over actions. Yet in order to ensure that local policies always have non-null probabilities on the local action space, we consider a regularized objective introduced in [17, 22]: for any given q-value table Q^i, let us define the mapping ϕ that returns the following local policy:

$$\phi(Q^i)(\cdot \mid s^i) = \underset{\rho \in \Omega(s^i)}{\arg\max} \left[\rho^T Q^i(s^i, \cdot) + \tau \nu^i_{s^i}(\rho)\right] \quad \forall s^i \in S_i \tag{3}$$

where $\tau > 0$ is a temperature parameter representing the weight given the regularization, and $\nu^i_{s^i}$ is a smooth and strongly concave function which takes infinite values outside of $\Omega(s^i)$. Strong concavity ensures the uniqueness of the solution $\phi(Q^i)$, and we call any local policy π^{*i} such that $\pi^{*i} = \phi(Q^{\pi^{-i}}_{\pi^{*i}})$ a smoothed best response to π^{-i}. If all agents follow a smoothed best-response, then the corresponding global policy is called an equilibrium.

Definition 2 (Equilibrium)**.** *A global policy π^* is an equilibrium iff every local policy π^i is a smoothed best response to other local policies π^{-i}*

To shorten the notation, we write $v'(Q^i, s^i, a^i)$ the expectation of the future gain as estimated by any table Q^i after taking action a^i in s^i:

$$v'(Q^i, s^i, a^i) = \sum_{s'^i} P^i(s^i, a^i, s'^i)[\phi(Q^i)(\cdot \mid s'^i)]^T Q^i(s'^i, \cdot) \tag{4}$$

Then writing $Q^{\pi^{-i}}_* = Q^{\pi^{-i}}_{\pi^{*i}}$, from lemma 1 we have that the equilibrium π^* and its associated q-functions $Q^{\pi^{*-i}}_*(s^i, a^i)$ are solutions to the following equations:

$$Q^{\pi^{*-i}}_* = \sum_s d^{\pi^*}(s \mid s^i) \sum_{a^{-i}} \pi^{*-i} r(s, a^i, a^{-i}) + \beta v'(Q^{\pi^{*-i}}_*, s^i, a^i)$$

for all $i \in \{1 \dots M\}$, $s^i \in S_i$, $a^i \in A_i$.

3 Weak-convergence of multi-scale Q-learning iterates

Let all agents maintain a local estimate $\hat{Q}^i$ of the q-function (1), and follow a local policy $\pi^i = \phi(\hat{Q}^i)$. The combined actions of all agents sample M local trajectories $\{(s^i_0, a^i_0, r^i_0), (s^i_1, a^i_1, r^i_1) \dots\}, i \in \{1 \dots M\}$. Let now all agents locally run a Q-learning update, so that each agent updates its local estimate $\hat{Q}^i_k$ at each timestep k:

$$\begin{aligned} \hat{Q}^i_{k+1}(s^i, a^i) = \hat{Q}^i_k(s^i, a^i) \\ + \alpha^i_k(s^i_k, a^i_k) \left[r_k + \beta[\phi(\hat{Q}_k)(s^i_{k+1})]^T \hat{Q}_k(s^i_{k+1}, \cdot) - \hat{Q}^i_k(s^i_k, a^i_k))\right] I_{k, s^i, a^i} \end{aligned} \tag{5}$$

with I_{k,s^i,a^i} the indicator of the event that the local state-action pair s^i, a^i is visited at timestep k. At this timestep, all other state-action pairs are therefore not updated, and the iterates are therefore asynchronous.

Note that in the original single-agent Q-learning, the collected reward $r(s, a)$ is exactly the expectation of the reward for the observed state-action pair (s, a). Here however, no agent ever collects a reward sampled according to the stationary distribution of the equilibrium policy as defined in the q-value eq. (1). In fact, no agent ever collects a reward sampled from any stationary distribution at all. Instead, we can notice that the collected reward depends on an unobserved Markovian global state process, and that the difference between the collected reward and the reward expected at equilibrium can be seen as a state-dependent noise. To treat this state-dependent noise, we can therefore apply results from the stochastic approximation theory concerning multi time scales iterates with Markovian noise. The full proof of this approach is detailed in the Appendice appendix A.2, and we call the resulting algorithm the *multi-scale Q-learning* algorithm. It allow us to reframe the convergence of the multi-scale Q-learning iterates as the problem of finding a specific order among agents, such that at equilibrium the change in policy of an agent of higher order creates only small perturbations among agents with a lower order.

In the next section we detail further what is means by agent ordering. Intuitively, for each agent we want to look at its best response dynamic, and identify a set of other agents such that this dynamic converges when all policies in the set are fixed. This will define a type of dependency between agents in the Dec-POMDP: if we can extract a total order on all agents from these dependencies, then it will suffice to assign learning rates following that order. Note that such a total order implies acyclic dependencies between agents. In Section 3.1, we will start by making explicit what is meant by ordering agents according to their dynamics through **A** 3.1. But such an assignment will force us to have as many learning rates as agents. Building further on the acyclic dependencies assumption, and to address a more concrete application, Section 3.2 zooms in on the case of the networked distributed POMDP (ND-POMDP), in which the shared reward is distributed among agents and the graph of connections between agents is known. We show knowledge about this graph can be exploited to reduce the number of different learning rates and build a faster algorithm.

3.1 Interaction structure between agents for multi-scale Q-learning

Consider a case in which agents are given learning rates such that every agent is learning at a different timescale. We start by defining precisely the total order needed on agents for this solution to converge.

Recall π, d^π as defined by (18), with $\phi(Q) \cdot d^\pi$ the corresponding stationary distribution over global state-action pairs. Therefore for any M-uplets Q there is an associated reward expectation taken over the stationary distribution of state-action pairs. We look at any agent i and its corresponding q-table Q^i. We denote $Q^{>i} = (Q^{i+1}, \dots, Q^M)$ and $Q^{<i} = (Q^1, \dots, Q^{i-1})$. Let us take a set of q-tables Q with its corresponding global policy $\pi = \phi(Q)$ such that

- For $j \leq i$, Q^j is any q-table in $S_j \times A_j$
- For $j > i$, Q^j is a q-table of the smoothed best response to π^{-j} as introduced in eq. (3). We write $Z^{\geq i+1}(Q^{<i+1})$ the $M - i$ q-tables $Q^{>i}$ thus defined.

Any disturbance to a local q-table $Q'^i \neq Q^i$ causes a corresponding change to $Z^{\geq i+1}(Q^{<i}, Q'^i)$. If the reward function is such that a local perturbation does not produce change in the reward expectation greater than the perturbation, then it will follow that the mean ODE approximating the local iterates (5) will have a single fixed point. We will now formalize this condition.

Assumption 3.1. *Let $Q'^i \in [-D, D]^{|S_i| \times |A_i|}$ be a local perturbation to Q^i within the constraint set. Write $Q' = (Q^{<i}, Q'^i, Z^{\geq i+1}(Q^{<i}, Q'^i))$ and $\pi' = \phi(Q')$. There exists an ordering of agents $\{1, \dots, M\}$ and $K \in (0, 1)$ such that for every agent i and its q-table Q^i, the reward function satisfies:*

$$\|R_\pi(s) - R_{\pi'}(s)\|_1 \leq K \|Q^i - Q'^i\|_\infty$$

Theorem 1. *Let us consider M agents locally updating their q-values estimates according to* (5) *with initial values $\hat{Q}_0^i \in [-D, D]$ for $D > 0$ such that $D > \frac{R}{\beta}$. Suppose that **A** 3.1 is satisfied with the ordering of agents $\{1, \dots, M\}$, and the learning rates $\{\alpha_i\}_{1 \dots M}$ follow **A** A.5 and **A** A.8, where*

α_i is the learning rate sequence of the i^{th} agent. If the discount factor β satisfies $\beta \leq 1 - K$, then the q-value estimates will converge weakly towards the smoothed best-response q-values Q^{*i}. Moreover, the deterministic global policy defined by s^i, $\pi*^i = \phi(Q^{*i})$ for all i is an equilibrium.

This learning rates attribution however forces us to have as many learning rates as we have agents. We notice that Section 3.1 defined a dependency between agent dynamics that can be represented by a directed acyclic graph (DAG). If such a dependency is known, then the graph can be used to assign a ranking to agents that allows for different agents to have the same learning rate sequence. We will now look at a specific class of Dec-POMDP with specific assumptions on agent interaction structure and see how this allows us to derive a faster algorithm.

3.2 Reward decomposition for multi-scale Q-learning

In this section we address further constraints on our Dec-POMDP that can relax the need for a total order on all agents. We now look at of Networked Distributed POMDPs (ND-POMDPs), a specific case of transition-independent Dec-POMDPs introduced by [18] to model distributed optimization problems like sensor network coordination. We now assume the shared reward can be written as a sum of M components $\{r^i\}_{1,\dots,M}$ such that for all $(s, a) \in S \times A$, $r(s, a) = \sum_{i=1}^{M} r^i(s^i, a^i, s^{U^i}, a^{U^i})$, where U^i is a subset of agents, and s^{U^i} - resp. a^{U^i} - is a vector concatenating local states and - resp. local actions - of agents in U^i. We say that agent i influences agent j if $i \in U^j$. Here, the total reward is not received by every agent, but rather distributed in the network that connects all agents.

Let the relationships between agents be modeled by a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ the set of vertices and $\mathcal{E}$ the set of edges, such that $|\mathcal{V}| = M$, and $(i \to j) \in \mathcal{E}$ iff agent i influences agent j. For every node i, we write $\mathcal{N}_{in}^i$ the set of nodes from which there is an edge to i in the graph, and $\mathcal{N}_{out}^i$ the set of nodes to which there is an edge from i in the graph. The neighborhood of node i is then noted $\mathcal{N}^i = \mathcal{N}_{in}^i \bigcup \mathcal{N}_{out}^i$. We write $\mathcal{NA}(i)$ the ancestors of i, that is the set of nodes for which there exists a path towards i. Similarly, we write $\mathcal{ND}(i)$ the descendants of i. Under **A** A.5, every agent learned at a different scale, for a total of M different scales. In ND-POMDPs, we can exploit the structure of the problem to attribute a smaller set of $\bar{M} \leq M$ scales to all agents.

We want to find a ranking function $rk : i \to rk(i) \in \{1, \dots, \bar{M}\}$, such that the proof of convergence of theorem 1 is preserved if every agent i is assigned the learning rate sequence $\alpha_k^{rk(i)}$. Let us start by rewriting **A** 3.1 as a loser, local assumption. To achieve this, first note that the only role of the total ordering in this assumption was to ensure that for every agent, the set of all other agents could be partitioned into two subsets: agents that need to learn slower and agents that need to learn faster. This was needed because in the general case, the dynamics of all iterates must be assumed to be dependent on all other iterates. Yet under our new DAG structure, we already know by construction that if the parents of i maintain fixed policies, then only a - possibly strict - subset of other agents will need to adapt their best responses to a change in the policy of agent i: its descendants and their respective ancestors. Therefore the convergence of the iterates for i can be ensured by a partition of other agents in 3 categories: some "faster" agents, some "slower" agents, and all other agents whose learning scale has no impact on the iterates. The possibility to gain in learning speed will depend on the size of that last subset. We can therefore rewrite:

Assumption 3.2. *For every agent i in $\mathcal{G}$ and with the same notations as **A** 3.1, there exists $K \in (0, 1)$ for the ordering of agents $\{\mathcal{NA}(i), i, \mathcal{ND}(i)\}$ such that $\|R_\pi(s) - R_{\pi'}(s)\|_1 \leq K\|Q^i - Q'^i\|_\infty$*

Then, any ranking that satisfies the following conditions will also preserve the convergence of theorem 5.

- (A) For any node i, nodes in $\mathcal{NA}(i)$ have a strictly inferior rank, and nodes in $\mathcal{ND}(i)$ have a strictly superior rank.
- (B) For any node i, there exists no two different nodes of the same rank in $\mathcal{NA}(i)$.

Let us now take any topological sorting algorithm and apply it to our directed acyclic graph: the total order on nodes it will return satisfies (A) by construction, and trivially satisfies (B) by giving a different rank to every node. Therefore it still returns $\bar{M} = M$ ranks. We give in appendix A.6 a straightforward attribution procedure for any DAG that returns $\bar{M} < M$ ranks whenever it is possible. An example of the application of that procedure to a real example can be found in Figure 2a.

As an example, let us consider a specific type of graph structures for which finding a ranking satisfying (A) and (B) is particularly trivial. We focus on the subset of graphs $\mathcal{T}$ defined the following way. First, it contains all trees. Secondly, for a given tree $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, it also contains any new graph $(\mathcal{V}, \mathcal{E} \cup \mathcal{E}')$, where $\mathcal{E}' \subset \{(i \to j) \mid (i,j) \in \mathcal{V},$ and $\exists$ path from i to j in $\mathcal{G}\}$. Then, finding a learning rate attribution that satisfies (A) and (B) is immediate: we only need attributing to every node the size of the longest path from a root - a node without incoming edge - to this node. We now show that under any learning rates attribution satisfying (A) and (B), convergence to a global equilibrium is preserved when every agent only receives a local reward gathered from its neighbors.

Theorem 2. *Let us consider M agents locally updating their q-values estimates according to iterates*

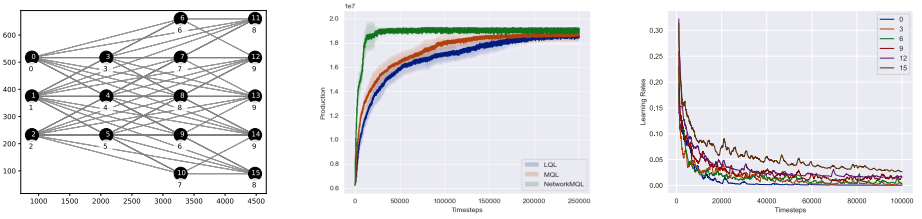
$$\hat{Q}^i_{k+1,c} = \hat{Q}^i_{k,c} + \alpha^i_k(s^i_k, a^i_k)\left[\bar{r}^i_k + \beta[\phi(\hat{Q}^i_k)(s^i_{k+1})]^T \hat{Q}^i_k(s^i_{k+1}, \cdot) - \hat{Q}^i_{k,c}\right] \tag{6}$$

*with $\bar{r}^i_k = \sum_{j \in \{i, \mathcal{N}^i\}} r^j(s^j_k, a^j_k, s^{U^j}_k, a^{U^j}_k)$, in the ND-POMDP with graph interaction network $\mathcal{G} \in \mathcal{T}$ satisfying **A** 3.2. Let the learning rates $\{\alpha_k\}$ be attributed by a ranking that satisifes (A) and (B). Then the conclusion of theorem 1 stands.*

The iterates eq. (6) will be labeled `NetworkMQL`. The proof is detailed in appendix A.7. We start by showing any ranking satisfying (A) and (B) preserves the convergence of theorem 5 and then that `TreeLRs` returns a learning rate distributions that belongs to that set.

4 Application to wind farm control

We validate the multi-scale approach on a case of a simulated wind farm with 16 turbines. An anonymized version of the code is available here: https://anonymous.4open.science/r/wfcrl-mql-D84F.



(a) Agent interaction DAG. (b) Evolution of power output (c) Evolution of learning rates

Figure 2: MQL: Multi-scale. NetworkMQL: Multi-scale with Reward Decomposition LQL: Local Q-learning. fig. 2a represents the 16 interacting wind turbines on a graph. The coordinates represent the location of each turbine in the farm. The levels used in the MQL algorithm are written in white, and the corresponding levels used in the NetworkMQL algorithm obtaines with Algorithm 1 are written in black. The total power output of the simulated 16 turbines wind farm averaged on 1000 time-steps is reported on Figure 2b. The evolution of learning rates under MQL based on scales attributed in Figure 2a is reported on Figure 2c for the first $100k$ time-steps.

Let us consider a farm of M wind turbines whose power output we want to maximize. In our multi-agent problem, every turbine is an agent. We assume that statistics on the wind inflow entering the farm can be represented by an irreducible and aperiodic Markovian process W taking values in a finite state space with transition kernel P_W. W is obviously not controllable by the agents. The production of each turbine i is a function of its yaw y^i, and of wind conditions statistics. This information can be gathered in its local state: we write S_i the finite local state space for agent i, and the finite global state space is $S = \times_i S_i$. The local action space A_i for agent i corresponds to the choice of increasing or decreasing its yaw by $1°$, or to let it unchanged, so that $A_i = \{-1, 0, +1\}$. The finite action space is similarly defined $A = \times_i A_i$. The reward $r(s, a)$ returns the total production of the farm after agents have picked action a in state s. Note that if agents are allowed to observe their local wind conditions, the problem is not transition-independent: any action taken by an agent can change the wind conditions at other agent's locations. This can be fixed by using a direct observation of W as wind statistics in the local state.

The transition function is $P = P_y P_W = \Pi_{i=1}^{M} P_y^i P_W$, where P_y^i is the transition kernel on the local yaw. Note that P_y^i is then entirely deterministic as for any $s^i, a^i, s'^i \in S_i \times A_i \times S_i$ we have $P(s^i, a^i, s'^i) = I\{s'^i = s^i + a^i\}$. It is easy to see that if all local policies are forced to maintain non-null probabilities on all local actions, then the local state processes will be irreducible and aperiodic.

A DAG modeling interactions between agents can be built the following way: from M nodes representing the M agents, we add an edge from $i \to j$ if turbine j is in the wake of turbine i. The reward can then be rewritten as a sum of local components $r(s,a) = \sum_i^M r^i(s^i, a^i, s^{U^i}, a^{U^i})$, where each r^i returns the production of agent i, and U^i is the set of in-neighbors of turbine i. We start by defining M learning sequences: for each rank in $\{1, \dots, \bar{M}\}$, let $0 < l_{\bar{M}} < \dots < l_1 < 1$ and the corresponding learning rate sequences be

$$\alpha_{k,c}^{l_i} = \frac{g}{n_k((s^i, a^i)_c)^{l_i}}$$

with $g > 0$ a gain and $n_k((s^i, a^i)_c) = \#$ visits to the cth state-action pair $(s^i, a^i)_c$ up to k. These sequences are standard for Q-learning algorithms. For our multiscale experiments, we modify the sequences by adding a term dependent on the time between visits, so that the final learning rate sequences are:

$$\alpha_{k,c}'^{l_i} = g\left(\frac{1}{n_k((s^i, a^i)_c)} + \frac{\log(T_{k,c}^i) - \log(T_{k-1,c}^i)}{T_{k,c}^i - T_{k-1,c}^i}\right)^{l_i}$$

where $T_{k,c}^i$ is the real time of the kth update to component c. This second term is motivated by the theory of convergence of asynchronous iterates - detailed in Appendix in **A** A.8: used alone, it ensures that learning happens at the same time-scale for all components of a single q-table. The first term used alone, or $\alpha_{k,c}^{l_i}$, ensures on the other hand that different agents learn at different time-scales. An example of the evolution of the resulting sequences of learning rates for Algortihm *MQL* can be found on Figure 2c, and we can see empirically that they preserve both properties. We use the same gain $g = 2$ for all algorithms. We run both Algorithm *MQL* eq. (5) and Algorithm *NetworkMQL* eq. (6) on a simulation of a wind farm with 16 wind turbines on 10 different seeds. We report the average production and standard deviation on Figure 2b. For *MQL*, we simply assign a different rank to every agent following a topological sort and use the M multi-scale learning rate sequences $\alpha_{k,c}'^{l_i}$. We compare with a naive Local Q-learning approach, where the standard Q-learning algorithm is run at every agent with the standard learning rates sequences $\alpha_{k,c}^{l_i}$. All agents are then given the fastest learning rate sequence corresponding to $l_i = 1$. For *NetworkMQL*, we use the procedure described in appendix A.6 to assign $\bar{M} \leq M$ ranks to all agents in the DAG. We obtain $\bar{M} = 9$ different ranks shown in fig. 2a and use the last 9 learning rate sequences in $\{l_i\}_{i \in 1 \dots M}$.

5 Conclusion

By allowing all agents to run a single-agent reinforcement algorithm in parallel, independent learning provides the simplest way to adapt these algorithms to cooperative multi agent environments. Although this approach has encountered experimental successes, it has no underlying theoretical guarantee. Inspired by the surprising success of such an approach on the wind farm control problem, we have highlighted a specific subclass of cooperative MARL problems: transition-independent Dec-PODMP where agent dynamics can be represented by a DAG. We show that in these problems the partial observability of the global state can be modeled as a Markovian perturbation in stochastic approximation iterates. We show that when there is an acyclic dependence structure between agent dynamics in these cooperative systems, a careful assignment of learning rate sequences following a multi-scale approach can be sufficient to establish convergence. In particular, knowledge of the interaction graph between agents in ND-POMDP can be exploited to assign learning rates to preserve convergence. We show how these results can be applied to wind farm control, a real optimization problem from the industry. Further work could extend these conclusions to systems with noisy local observations or non-independent transition functions. Furthermore, independent learning has often encountered experimental success without the multiscale approach in multiagent reinforcement learning settings ([29, 26]), and our acyclic dependence analysis could provide a basis to find a theoretical explanation of these results.

References

[1] Cristina L. Archer et al. "Review and evaluation of wake loss models for wind energy applications". In: *Applied Energy* 226 (Sept. 2018), pp. 1187–1207.

[2] Nolan Bard et al. "The Hanabi Challenge: A New Frontier for AI Research". In: *Artif. Intell.* 280 (2019), p. 103216.

[3] Lucas Baudin and Rida Laraki. "Fictitious play and best-response dynamics in identical interest and zero-sum stochastic games". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 1664–1690.

[4] Raphen Becker et al. "Solving Transition Independent Decentralized Markov Decision Processes". In: *J. Artif. Intell. Res.* 22 (2004), pp. 423–455.

[5] Daniel S. Bernstein, Shlomo Zilberstein, and Neil Immerman. "The Complexity of Decentralized Control of Markov Decision Processes". In: *ArXiv* abs/1301.3836 (2000).

[6] Claire Bizon Monroc et al. "Actor Critic Agents for Wind Farm Control". In: *2023 American Control Conference (ACC)*. 2023, pp. 177–183. DOI: 10.23919/ACC55779.2023.10156453.

[7] Vivek S. Borkar. "Stochastic approximation with two time scales". In: *Systems & Control Letters* 29.5 (1997), pp. 291–294. ISSN: 0167-6911.

[8] Van-Hai Bui, Thai-Thanh Nguyen, and Hak-Man Kim. "Distributed operation of wind farm for maximizing output power: A multi-agent deep reinforcement learning approach". In: *IEEE Access* 8 (2020), pp. 173136–173146.

[9] Jilles S Dibangoye, Christopher Amato, and Arnoud Doniec. "Scaling up decentralized MDPs through heuristic search". In: *arXiv preprint arXiv:1210.4865* (2012).

[10] Claudia V Goldman and Shlomo Zilberstein. "Decentralized control of cooperative systems: Categorization and complexity analysis". In: *Journal of artificial intelligence research* 22 (2004), pp. 143–174.

[11] Peter Graf et al. "Distributed Reinforcement Learning with ADMM-RL". In: *2019 American Control Conference (ACC)*. 2019, pp. 4159–4166. DOI: 10.23919/ACC.2019.8814892.

[12] Tommi Jaakkola, Michael Jordan, and Satinder Singh. "On the Convergence of Stochastic Iterative Dynamic Programming Algorithms". In: *Neural Computation* 6 (Nov. 1994), pp. 1185–1201.

[13] Elie Kadoche et al. "Marlyc: Multi-agent reinforcement learning yaw control". In: *Renewable Energy* 217 (2023), p. 119129.

[14] Ali C. Kheirabadi and Ryozo Nagamune. "A quantitative review of wind farm control with the objective of wind farm power maximization". In: *Journal of Wind Engineering and Industrial Aerodynamics* 192 (2019), pp. 45–73. ISSN: 0167-6105. DOI: https://doi.org/10.1016/j.jweia.2019.06.015. URL: https://www.sciencedirect.com/science/article/pii/S0167610519305240.

[15] Harold J. Kushner and G. George Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Stochastic modelling and applied probability. Springer New York, NY, 2003.

[16] David S Leslie, Steven Perkins, and Zibo Xu. "Best-response dynamics in zero-sum stochastic games". In: *Journal of Economic Theory* 189 (2020), p. 105095.

[17] David S. Leslie and E. J. Collins. "Convergent Multiple-Timescales Reinforcement Learning Algorithms in Normal Form Games". In: *The Annals of Applied Probability* 13.4 (2003), pp. 1231–1251. ISSN: 10505164.

[18] Ranjit Nair et al. "Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs". In: *AAAI*. Vol. 5. 2005, pp. 133–139.

[19] Hadi Nekoei et al. "Dealing with non-stationarity in decentralized cooperative multi-agent deep reinforcement learning via multi-timescale learning". In: *Conference on Lifelong Learning Agents*. PMLR. 2023, pp. 376–398.

[20] Afshin Oroojlooyjadid and Davood Hajinezhad. "A review of cooperative multi-agent deep reinforcement learning". In: *Applied Intelligence* 53 (2019), pp. 13677–13722.

[21] Jeffrey S. Rosenthal. "Convergence Rates for Markov Chains". In: *SIAM Review* 37.3 (1995), pp. 387–405. ISSN: 00361445.

[22] Muhammed Sayin et al. "Decentralized Q-learning in zero-sum Markov games". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 18320–18334.

[23] Muhammed O Sayin, Francesca Parise, and Asuman Ozdaglar. “Fictitious play in zero-sum stochastic games”. In: *SIAM Journal on Control and Optimization* 60.4 (2022), pp. 2095–2114.

[24] Paul Stanfel et al. “Proof-of-concept of a reinforcement learning framework for wind farm energy capture maximization in time-varying wind”. In: *Journal of Renewable and Sustainable Energy* 13.4 (Aug. 2021).

[25] Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018.

[26] Ming Tan. “Multi-Agent Reinforcement Learning: Independent versus Cooperative Agents”. In: *International Conference on Machine Learning*. 1997.

[27] John N. Tsitsiklis. “Asynchronous Stochastic Approximation and Q-Learning”. In: *Mach. Learn.* 16.3 (Sept. 1994), pp. 185–202. ISSN: 0885-6125.

[28] Christopher Watkins and Peter Dayan. “Technical Note: Q-Learning”. In: *Machine Learning* 8 (May 1992), pp. 279–292.

[29] C. S. D. Witt et al. “Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge?” In: *ArXiv* abs/2011.09533 (2020).

[30] Zhiwei Xu et al. “Model-Free Optimization Scheme for Efficiency Improvement of Wind Farm Using Decentralized Reinforcement Learning”. In: *IFAC-PapersOnLine* 53.2 (2020). 21st IFAC World Congress, pp. 12103–12108. ISSN: 2405-8963.

[31] K. Zhang, Zhuoran Yang, and Tamer Başar. “Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms”. In: *ArXiv* abs/1911.10635 (2019).

[32] Kecheng Zhang et al. “Multi-Vehicle Routing Problems with Soft Time Windows: A Multi-Agent Reinforcement Learning Approach”. In: *ArXiv* abs/2002.05513 (2020).

A Appendix / supplemental material

A.1 Proof of Recursive form of local q-function

Proof of Lemma lemma 1 We can rewrite (1) as

$$Q_{\pi^i}^{\pi^{-i}}(s^i, a^i) \tag{7}$$

$$= \sum_s d^\pi(s \mid s^i) \mathbb{E}_{a_0 \sim (\pi^i, \pi^{-i})} \left[r(s_0, a_0) \mid s_0^i = s^i, a_0^i = a^i, s_0 = s \right] \tag{8}$$

$$+ \beta \sum_s d^\pi(s \mid s^i) \mathbb{E}_{a_k \sim (\pi^i, \pi^{-i}), s_k \sim P} \left[\sum_{k=1}^{\infty} \beta^{k-1} r(s_k, a_k) \mid s_0^i = s^i, a_0^i = a^i, s_0 = s \right] \tag{9}$$

While the term in (8) can be expressed as $\sum_s d^\pi(s \mid s^i) \sum_{a^{-i}} \pi^{-i}(a^{-i}|s) r(s, a^i, a^{-i})$, the term in (9) can be developed in the following way

$$\begin{aligned}
&\beta \sum_s d^\pi(s \mid s^i) \mathbb{E}_{s_1, a_1^i} \left[\mathbb{E}_{a_k \sim \pi, s_k \sim P} \left[\sum_{k=1}^{\infty} \beta^{k-1} r(s_k, a_k) \mid s_0 = s, a_0^i = a^i, s_1 = s', a_1^i = a'^i \right] \middle| s_0 = s, a_0^i = a^i \right] \\
&= \beta \sum_s d^\pi(s \mid s^i) \sum_{a^{-i}} \pi^{-i}(a^{-i} \mid s) \sum_{s'} P(s, a, s') \sum_{a'^i} \pi^i(a'^i \mid s'(i)) \times \\
&\quad \mathbb{E}_{a_k \sim \pi, s_k \sim P} \left[\sum_{k=1}^{\infty} \beta^{k-1} r(s_k, a_k) \mid s_1 = s', a_1^i = a'^i \right] \\
&= \beta \sum_{s'^i} P^i(s^i, a^i, s'^i) \sum_{a'^i} \pi^i(a'^i \mid s'(i)) \left[\sum_s d^\pi(s \mid s^i) \sum_{a^{-i}} \pi^{-i}(a^{-i} \mid s) \sum_{s'} P^{-i}(s^{-i}, a^{-i}, s'^{-i}) \times \right. \\
&\quad \left. \mathbb{E}_{a_k \sim \pi, s_k \sim P} \left[\sum_{k=1}^{\infty} \beta^{k-1} r(s_k, a_k) \mid s_1^i = s'^i, s_1^{-i} = s'^{-i}, a_1^i = a'^i \right] \right] \\
&= \beta \sum_{s'^i} P^i(s^i, a^i, s'^i) \sum_{a'^i} \pi^i(a'^i \mid s'(i)) \left[\mathbb{E}_{s_1 \sim d^\pi, a_k \sim \pi, s_k \sim P} \left[\sum_{k=1}^{\infty} \beta^{k-1} r(s_k, a_k) \mid s_1^i = s'^i, a_1^i = a'^i \right] \right] \\
&= \beta \sum_{s'^i} P^i(s^i, a^i, s'^i) \sum_{a'^i} \pi^i(a'^i \mid s'(i)) Q_{\pi^i}^{\pi^{-i}}(s'^i, a'^i)
\end{aligned}$$

where the second line to the last is due to the definition of the stationary distribution.

A.2 Conditions for the weak convergence of synchronous multi-scale iterates with Markovian noise

Weak convergence of stochastic approximation for two time-scales systems were proven in [15]. We formally extend these results to the multi-scale case. We consider the constrained case: at each iteration, the iterates are projected on a defined admissible space H. We assume that H is a hyperrectangle $H = [h_1, b_1] \times [h_2, b_2] \times \cdots \times [h_d, b_d]$ with $(h_i, b_i) \in \mathbb{R}^2$ for $i \in \{1, \ldots, d\}$ and $d > 0$ the dimension of the iterates. The operator Π_H is used to denote this projection on H.

Consider M interdependent stochastic approximation processes $\theta_k^1, \ldots, \theta_k^M$ updated according to iterates:

$$\theta_{k+1}^i = \Pi_H \left[\theta_k^i + \alpha_k^i Y_k^i \right] = \theta_k^i + \alpha_k^i (F^i(\theta_k, \xi_k^i) + \delta U_k^i) + B_k^i \tag{10}$$

where $\theta_k = (\theta_k^1, \cdots, \theta_k^M)$, $\{\xi_k^i\}$ are noise sequences, $F^i(\cdot, \cdot)$ are functions of θ and ξ^i, $\delta U_{k+1}^i = Y_k^i - F^i(\theta_k, , \xi_k^i)$ are martingale noise differences, $\alpha_k^i := \alpha^i(k) > 0$ are learning rates for timescale i at iterate k, and B_k^i is a correction term to project the iterate on H, henceforth referred as reflection terms.

Let $\{\mathcal{F}_k\}$ be a sequence of non-decreasing $\sigma-$algebra generated by $\{\theta_j^i, Y_{j-1}^i, \xi_j^i, j \leq k, i \leq M\}$, and $\mathbb{E}_k$ refers to the associated conditional expectation $\mathbb{E}[\cdot|\mathcal{F}_k]$, and we have $\mathbb{E}_k Y_k^i = F^i(\theta_k, \xi_k^i)$. To be concise, we will use the notations

$$\theta^{<i} := (\theta^1, \ldots, \theta^{i-1}), \quad \theta^{\geq i} := (\theta^i, \ldots, \theta^M).$$

We now lay down the assumptions needed to ensure convergence. Let Ξ be a complete and separable metric space, and A be an arbitrary compact set in Ξ. We start by standard assumptions for stochastic approximation algorithms: the sequences of observations Y_k^i are uniformly integrable, and at each timestep their expectations

are given by a continuous function of the iterate θ_k^i. The main idea is that an error term δU_k^i of null expectation will be averaged out through the iterations, so that as k goes to infinity, the behavior of the iterates can be described without the error terms. We make the following assumption for $i = 1, \cdots, M$.

We first start with basic assumptions from stochastic approximation theory:

Assumption A.1. *The $\{Y_k^i\}$ are uniformly integrable, and can be written $Y_k^i = F^i(\theta_k, \xi_k^i) + \delta U_k^i$ with $\{\delta U_k^i\}_k$ martingale noise differences $\mathbb{E}_k \delta U_k^i = 0$ and $F^i(\cdot, \xi^i)$ functions continuous in θ, and continuous in $\xi^i \in A$.*

Here, F^i is still dependent on the error sequences ξ_k^i whose expectations are not null. Yet, the Markovian property of these sequences, combined with a constraint on the rate of evolution of the learning rates (see **A** A.5), can be exploited to construct an approximation of $F^i(\cdot, \xi^i)$ that does not depend on ξ^i. We detail some assumptions on the Markovian noise processes that can be considered.

Assumption A.2. *The noise processes $\{\xi_k^i\}$ are bounded with values in Ξ, and Markovian: they admit a transition function $P^i(\cdot, \cdot|\theta)$ such that $P^i(\cdot, A|\theta)$ is measurable for each Borel set $A \subset \Xi$, and $P^i(\xi_{k+1}^i \in \cdot|\mathcal{F}_k) = P^i(\xi_k^i, \cdot|\theta_k)$. This transition function is continuous and does not depend on k. For any compact $A \in \Xi$ and $\mu \in (0, 1)$ such that, there exists a compact A' such that $P(\xi_{k+1}^i \in A' \mid \xi_k^i) \geq 1 - \mu$ for all $\xi_k^i \in A$.*

We now define the fixed $\theta-$chain $\{\xi_k(\theta)\}$, the Markov chain on state space Ξ with the fixed transition function $P(\xi, \cdot, |\theta)$. It is the noise process starting from k if θ stayed constant, i.e., $\{\xi_{k+j}(\theta), j \geq 0, \xi_k(\theta) = \xi_k)\}$. The continuous function of the actual noise process can be approximated by the continuous function of the fixed-chain process $F^i(\cdot, \xi_k^i)$ (see the proof of Theorem 8.4.3 [15], p.271-275) if the rate of change of the learning rates is slow enough. If we can construct a function $\hat{F}^i(\cdot)$ of θ that does not depend on the process ξ^i, such that $\hat{F}^i(\theta)$ is a local average of the $F^i(\cdot, \xi_k^i)$, then $\hat{F}^i(\theta_k)$ is also an approximation of $F^i(\theta_k, \xi^i)$ as $k \to \infty$. We detail these assumptions here:

Assumption A.3. *The set $\{F^i(\theta_k, \xi_k^i)\}_k$ is uniformly integrable. For any $j \geq 0$ with $\xi_j^i \in A$ the set $\{F^i(\theta, \xi_{j+k}^i(\theta))\}_{k \geq 0}$ is uniformly integrable, where $\xi_{j+k}^i(\theta)$ is the fixed-θ chain with initial conditions ξ_j^i and transition function $P^i(\xi, \cdot|\theta_j)$.*

Assumption A.4 (Averaging condition)**.** *There exists a continuous function $\bar{F}^i(\cdot)$ such that for each θ and on any compact set $A \in \Xi$*

$$\lim_{(k,m)\to\infty} \frac{1}{m} \sum_{j=k}^{k+m-1} \mathbb{E}_k \left[F^i(\theta, \xi^i(\theta)) - \bar{F}^i(\theta) \right] I_{\xi_k^i \in A} = 0$$

We establish a different timescale to correspond to each process. For $i, j \in \{1, \cdots, M\}$, let $t_k^j = \sum_{l=0}^{k-1} \alpha_l^j$, and $\theta_{\alpha^j}^{i,0}(t)$ be the piecewise interpolation of the process θ_k^i on the j-th timescale defined as

$$\theta_{\alpha^j}^{i,0}(t) = \theta_0^i, \ t \leq 0, \quad \theta_{\alpha^j}^{i,0}(t) = \theta_k^i, \ t \in [t_k^j, t_{k+1}^j]$$

Then, the shifted continuous time interpolation $\theta_{\alpha^j}^{i,k}(\cdot)$ is simply the interpolation "started" from a specific time-step k:

$$\theta_{\alpha^j}^{i,k}(t) = \theta_{\alpha^j}^{i,0}(t_k^j + t) \tag{11}$$

and we let $m^{(j)}(t) = \{\kappa : t_\kappa^j \leq t \leq t_{\kappa+1}^j\}$. Similarly, we define $B_{\alpha^j}^{i,k}$ the shifted continuous time interpolation at the j-th timescale of the sequence of reflection terms B_k^i. We are interested in the behavior of $\theta_{\alpha^j}^{i,k}(\cdot)$ and $B_{\alpha^j}^{i,k}(\cdot)$ as $t_k^j \to \infty$ while $\alpha_k^j \to 0$.

We now lay out further constraints on the learning rate sequences. The first two are standard for the stochastic approximating literature: intuitively they require the learning rates to go towards zero, but not too quickly. The third assumption is what makes the iterates multi-scale: it imposes a hierarchy between the M sequences that ensures every iterate is learning at a different timescale.

Assumption A.5 (Assumption on learning rates)**.** *For each $i \in \{1, \cdots, M\}$,*

- *(Classical rates) $\lim_k \alpha_k^i = 0$ and $\sum_{k=0}^{\infty} \alpha_k^i = \infty$*
- *(Slow changes) there is a sequence of integers $a_n^i \to \infty$ such that*

$$\lim_n \sup_{0 \leq j \leq a_n^i} \left| \frac{\alpha_{n+j}^i}{\alpha_n^i} - 1 \right| = 0$$

- *(Multi-scale) $\frac{\alpha_k^i}{\alpha_k^j} \to 0$, as $k \to \infty$, whenever each $i < j$.*

With the expectations $\mathbb{E}_k Y_k^i$ being approximated by $\bar{F}^i(\theta_k)$ as k goes to ∞, the interpolations of the iterates $\theta_{\alpha^j}^{i,k}$ will be shown to admit limit processes following mean ODEs defined by the $\bar{F}^i$. The solution of the ODE can then be used to characterize the asymptotic properties of the θ_k^i for $i = 1, \cdots, M$. Thanks to the multi-scale assumption, at any timescale j the interpolation for all iterates learning at a slower timescale $i < j$ will follow the null ODE. Intuitively, they evolve so slowly that they can be considered constant at the j-th timescale. Similarly, the interpolations for all iterates learning at a faster timescale can be considered to have reached the limit of their respective mean ODE, if it exists. We consider the case where the ODE for every limit process for any timescale has a unique asymptotically stable point.

Assumption A.6. *There exists a continuous function $\zeta^i(\theta^{<i})$ such that, for any set of initial conditions θ, the solution to the following ODE has a unique asymptotically stable point $(\theta^{<i}, \zeta^i(\theta^{<i}))$ for $i \geq 2$:*

$$\begin{aligned} \dot{X}^j &= 0 \quad \text{for} \quad j < i \\ \dot{X}^i &= \bar{F}^i(X^{<i+1}, Z^{\geq i+1}(X^{<i+1})) + b^i. \end{aligned}$$

where b^i is the reflection on H, and

$$Z^{\geq i}(\theta_k^{<i}) = (\zeta^i(\theta_k^{<i}), Z^{\geq i+1}(\theta_k^{<i}, \zeta^i(\theta_k^{<i}))), \quad i = 2, \cdots, M-2 \tag{12}$$

with $Z^{\geq M-1}(\theta_k^{<M-1}) = (\zeta^{M-1}(\theta_k^{<M-1}), \zeta^M(\theta_k^{<M-1}, \zeta^{M-1}(\theta_k^{<M-1})))$.

When applying our multi-scale iterates to our Dec-POMDP problem, this assumption will enforce strong constraints on the dynamics of the multi-agent system. In Section 3, we will introduce specific DAG structures on agent interaction that can satisfy them, and a concrete example will be given in Section 4.

Note that the reflection terms b^i of the projected ODE must live within a convex space $\Upsilon(X^i)$, defined the following way: on the interior of H, $\Upsilon(X^i) = \{0\}$, the set only containing the null vector, and on the boundary of H, $\Upsilon(X^i)$ is the infinite convex cone generated by the outer normals at X^i of the faces on H on which X^i lies.

Now we state the week convergence of the iterates (10) in the following theorem.

Theorem 3 (Weak convergence of multi-scale iterates with Markovian noise)**.** *Consider iterates* (10)*. Let $\{\theta_{\alpha^j}^{i,k}(\cdot)\}$ be the interpolation of the process θ_k^i on the j-th timescale, defined by* (11)*. If **A** A.1 to A.5 hold, then $\{\theta_{\alpha^1}^{1,k}(\cdot)\}$ admits a subsequence which converges towards a process $\theta^1(\cdot)$ such that:*

$$\dot{\theta^1} = \bar{F}^1(\theta^1, Z^{\geq 2}(\theta^1)) + b^1, \quad b^1(t) \in -\Upsilon(\theta^1(t)) \tag{13}$$

where b^1 is the reflection, that is the minimum force needed to keep θ^1 in H. Moreover, for any $\delta > 0$, the fraction of time spent by $\theta^1(\cdot)$ in any $\delta-$neighborhood around the set of limit points of (13) *on the interval $[0, T]$ goes to one in probability as $T \to \infty$.*

theorem 3 is a straightforward extension of the week convergence result established for two timescale iterates by [Theorem 8.6.1, [15], p.286] to the case of M time scales. An example of the extension procedure from two-scale to multi-scale can be found in [17]. The idea behind Kushner's original proof in [15] for the two-timescale case is that the noise induced by the Markovian sequences $\{\xi_k^i\}$ can be seen as perturbations to local averages defined by the functions $\bar{F}^i$. This allows to approximate the iterates in continuous time by a projected ODE.

A.3 Extension to asynchronous iterates

We will now consider the case where the iterates are updated asynchronously: that is, not all elements of the θ^i are updated at every iteration.

We index all elements in every θ^i by $c \in \{1 \dots C\}$, and the C elements are updated in an asynchronous manner. Let $\alpha_{k,c}^i$ be the learning rate for element c of iterate i at timestep k: all elements within a single iterate are given the same sequences of learning rates, so that we use the notation $\alpha_k^i = \alpha_{k,1}^i = \alpha_{k,2}^i = \cdots = \alpha_{k,C}^i$. The M iterates in (10) can therefore be seen as $M \times C$ iterates, with the updates to each component following:

$$\theta_{k+1,c}^i = \Pi_H \left[\theta_{k,c}^i + \alpha_{k,c}^i Y_{k,c}^i \right] \tag{14}$$

The time between the kth and $(k+1)$th updates of the element indexed by c in $\{\theta_k^i\}_k$ is given by the random variable $\tau_{k,c}^i$. Because the kth update can happen at a different time for two components, we need another timeline to analyze the behavior of the iterates. We will look at their behaviors in the "real time", so that the kth update at element c in the $\{\theta_k^i\}_k$ is done at the real time $T_{k,c}^i = \sum_{n=0}^{k-1} \tau_{n,c}^i$. We note $\Gamma_{k,c}^i = \sum_{n=0}^{k-1} \alpha_n^i \tau_{n,c}^i$ the corresponding scaled real time, and introduce the real-time interpolation $\hat{\theta}_c^i$: $\hat{\theta}_c^i(t) = \theta_{k,c}^i$ on $[T_{k,c}^i, T_{k+1,c}^i)$.

Like in (11), we look at the shifted piecewise constant interpolations θ^i_{c,α^j} of the sequences $\{\theta^i_{k,c}\}_k$ at every timestep $j = \{1, \dots, C\}$ in the iterate time, that is the continuous interpolations whose origins are at any arbitrary timestep k. Here again, since all components do not reach a given timestep at the same time, we define the shifted interpolates as starting at arbitrary real times υ. For this purpose, we introduce functions $p^i_c(\upsilon)$, that return the index of the first update at an element c of iterate i after a given real time υ:

$$p^i_c(\upsilon) = \min\left\{k : \sum_{n=0}^{k-1} \tau^i_{n,c} \geq \upsilon\right\}, \quad \forall \upsilon > 0,$$

The shifted interpolates are then

$$\theta^{i,\upsilon}_{c,\alpha^j}(t) = \theta^i_{k+p^i_c,c}, \quad t \in [t^{ij,\upsilon}_{k,c}, t^{ij,\upsilon}_{k+1,c}), \qquad t^{ij,\upsilon}_{k,c} = \sum_{n=p^i_c(\upsilon)}^{k-1} \alpha^j_n \tag{15}$$

and the shifted real-time interpolations $\theta^{i,\upsilon}_{c,\alpha^j}(\cdot)$ are defined similarly:

$$\hat{\theta}^{i,\upsilon}_{c,\alpha^j}(t) = \theta^i_{k,c}, \quad t \in [\Gamma^{ij,\upsilon}_{k+p^i_c,c}, \Gamma^{ij,\upsilon}_{k+1,c}) \qquad \Gamma^{ij,\upsilon}_{k,c} = \sum_{n=p^i_c(\upsilon)}^{k-1} \alpha^j_n \tau^i_{n,c} \tag{16}$$

We now extend the definitions of the $\sigma-$algebra used in Appendix A.2. Two sets of random variables need to be considered at every iteration: the $Y^i_{k,c}$ and the $\tau^i_{k+1,c}$. The corresponding $\sigma-$algebras should measure all variables observed in the "past" up to the relevant moment during update $k+1$. Again reasoning in real time, note that update $k+1$ is made after having observed Y^i_k, but before entering the next waiting time $\tau^i_{k+1,c}$. This corresponds to two slightly different sequences of $\sigma-$algebras:

$$\mathcal{F}^{i,\tau}_{k,c} = \{\theta^i_{0,c}, Y^i_{j-1,h}, \xi^i_{j-1,h}, \tau^i_{j-1,h} \mid T^i_{j,h} \leq T^i_{k+1,c}\}$$
$$\mathcal{F}^{i,Y}_{k,c} = \{\theta^i_{0,c}, Y^i_{j-1,h}, \xi^i_{j-1,h} \mid T^i_{j,h} < T^i_{k+1,c}\} \bigcup \{\tau^i_{j-1,h} \mid T^i_{j,h} \leq T^i_{k+1,c}\}$$

We write the associated conditional expectations $\mathbb{E}^{i,\tau}_{k,c}$ and $\mathbb{E}^{i,Y}_{k,c}$.

Let us denote the component-wise error sequences $\xi^i_{k,c}$ $\delta U^i_{k,c}$, $\xi^i_k = (\xi^i_{k,1}, \dots, \xi^i_{k,C})$, and $\delta U^i_k = (\delta U^i_{k,1}, \dots, \delta U^i_{k,C})$. We assume **A** A.1 to A.5 hold, with any statement on a sequence X^i_k interpreted as holding for all component-wise sequences $X^i_{k,c}$. We make the additional assumptions on the time intervals between updates:

Assumption A.7. *For all i, the sequence of intervals between updates $\{\tau^i_{k,c}\}_k$ is uniformly integrable, and there exists $\bar{u}^i_c \geq 1$ such that the $\mathbb{E}^{i,\tau}_{k+1,c}\left[\tau^i_{k,c}\right]$ are in the bounded interval $\left[1, \bar{u}^i_c\right]$ uniformly in k.*

Assumption A.8. *Every component's learning rate $\alpha^i_{k,c}$ can be written as a local average of positive real-valued functions f^i:*

$$\alpha^i_{k,c} = \frac{1}{\tau_{k,c}} \int_{T^i_{k,c}}^{T^i_{k,c}+\tau^i_{k,c}} f^i(s)ds \quad \textit{such that} \int_0^\infty f^i(s)ds = \infty \textit{ and } \lim_{s\to\infty} f^i(s) = 0$$

Assumption A.9. *There exists a continuous function $\zeta^i(\theta^{<i})$ such that, for any set of initial conditions θ, the solution to the following ODE has a unique asymptotically stable point $(\theta^{<i}, \zeta^i(\theta^{<i}))$ for $i \geq 2$:*

$$\dot{X}^j = 0 \quad \text{for} \quad j < i$$
$$\dot{X}^i = \frac{\bar{F}^i(X^{<i+1}, Z^{\geq i+1}(X^{<i+1}))}{u^i_c} + \hat{b}^i.$$

with $u^i_c(t)$ with values in $[1, \bar{u}^i_c]$, $\hat{b}^i$ the term of projection on H, the Z^i have been defined in (12)

Then, we can state the weak convergence result for the asynchronous multi-scale iterates.

Theorem 4 (Weak convergence of asynchronous multi-scale iterates with Markovian noise)**.** *Consider iterates* (14)*, updated asynchronously following the time interval sequences $\{\tau^i_{k,c}\}_k$. If Assumptions **A** A.1 to A.5 hold, and Assumptions **A** A.7 to A.9 also hold, then the conclusion of theorem 3 still holds with the limit process:*

$$\dot{\hat{\theta}}^1_{c,\alpha^1}(t) = \frac{\bar{F}^1(\theta^1_{c,\alpha^1}(t), Z^{\geq 2}(\theta^1_{c,\alpha^1})(t))}{u^i_c(t)} + \hat{b}^1_{c,\alpha^1}(t) \qquad u^i_c(t) \in [1, \bar{u}^i_c]. \tag{17}$$

The weak convergence of asynchronous updates for the single-agent case has been established in [Theorem 12.3.5 [15]], and this is an extension to the multi-scale case. As previously for theorem 3, the extension is

derived by writing the ODEs for the continuous approximations at all iterate timescales. Unlike before, the ODEs are now dependent on the continuous approximation at the real timescale. A simple relation between the approximations at real and iterate timescales introduced by [15] can then be used to derive ODEs for the latter and conclude the proof.

With the weak convergence of the multi-scale iterates laid out, we are now ready too apply these results to our multi-scale Q-learning iterates eq. (5).

A.4 Application to multi-scale Q-learning iterates

Theorem 5. *Consider the multi-scale Q-learning iterates* (5)*. If* **A** *A.5, A.8 and A.9 are true, let all* $\hat{Q}^i_{0,c} \in [-D, D]$ *for* $D > 0$ *such that* $D > \frac{R}{\beta}$ *with* R *the reward bound, then the conclusions of theorem 4 hold.*

Proof. Let us consider any local state action pair s^i, a^i of any iterate Q^i. By assumption on the transition kernel **A** 2.2 and the design of the mapping ϕ, the sequence of times between two visits are uniformly integrable. All return times must moreover be at least 1. The $\mathbb{E}_{n+1,c}\tau^i_{n,c}$ are therefore uniformly bounded with values in an interval $[1, u^i_c]$ with $u^i_c \geq 1$, therefore satisfying **A** A.7.

In the following, we write $\{\bar{F}^i_c(Q)\}_c$ $\mathbb{R}$-valued continuous functions for $c \in \{1 \dots, |S_i||A_i|$, with ϕ defined in eq. (3) for any update to a component c we write:

$$\pi_k = (\pi^1_k, \dots, \pi^M_k) \quad \pi^j_k = \phi(\hat{Q}^j_k) \quad \hat{Q}_k = (\hat{Q}^1_k, \dots, \hat{Q}^M_k) \quad \hat{Q}^i_k = \{\hat{Q}^i_{k,c}\}_{c=1,\dots,C} \tag{18}$$

We can rewrite the iterates (5) in real time in the following way:

$$\begin{aligned}
\hat{Q}^i_{k+1,c} &= \hat{Q}^i_{k,c} + \alpha^i_{k,c} I_{k,c} \left[\bar{F}^i_c(\hat{Q}_k) + \delta U^i_{k,c} + \xi^i_{k,c}\right] \\
\delta U^i_{k,c} &:= Y^i_{k,c} - \mathbb{E}^Y_{k,c}[Y^i_{k,c}] \\
\xi^i_{k,c} &:= \mathbb{E}^Y_{k,c}[Y^i_{k,c}] - \bar{F}^i_c(\hat{Q}^i_k) \\
Y^i_{k,c} &:= r_k + \beta[\phi(\hat{Q}^i_k)(\cdot \mid s^i_{k+1})]^T \hat{Q}^i_k(s^i_{k+1}, \cdot) - \hat{Q}^i_{k,c} \\
&= F^i_c(\hat{Q}_k, \xi^i_{k,c}) + \delta U^i_{k,c}
\end{aligned} \tag{19}$$

where

$$\begin{aligned}
\bar{F}^i_c(Q) &= \sum_s d^{\pi_k}(s \mid s^i) \sum_{a^{-i}} \pi^{-i}_k(a^{-i}|s) r(s, a^i, a^{-i}) + \beta v'^i(Q^i, (s^i, a^i)_c) - Q^i_c \\
F^i_c(Q, \xi) &= \bar{F}^i_c(Q) + \xi
\end{aligned}$$

with $v'^i(Q, (s^i, a^i)_c) = v'^i(Q, s^i, a^i)$ for s^i, a^i the cth component of Q^i and recall that

$$v'(Q^i, s^i, a^i) = \sum_{s'^i} P^i(s^i, a^i, s'^i)[\phi(Q^i)(\cdot \mid s'^i)]^T Q^i(s'^i, \cdot)$$

We will now show that the iterates eq. (19) are in fact equivalent to their constrained version:

$$\hat{Q}^i_{k+1,c} = \Pi_{[-D,D]}\left(\hat{Q}^i_{k,c} + \alpha^i_{k,c} I_{k,c}\left[\bar{F}^i_c(Q^i_{k,c}) + \delta U^i_{k,c} + \xi^i_{k,c}\right]\right)$$

Indeed for any i, k, c, we have $\alpha^i_{k,c} \in (0, 1)$. Per definition of the discount factor, it is also true that $\beta \in (0, 1)$. It follows that since $\hat{Q}^i_{0,c} \in [-D, D]$ for all c and $R < \beta D$, and $\phi(\hat{Q}^i_k)$ is a probability distribution, then we have $\sup_k ||\hat{Q}^i_{k,c}|| < D$ for all c and the iterates will never leave the hyper-rectangle defined by $[-D, D]^{|S_i||A_i|}$. This means that for constrained iterates with constraint space $[-D, D]^{|S_i||A_i|}$, the induced reflexion term will always equal zero.

The $Y^i_{k,c}$ are then uniformly bounded, and the F^i_c are moreover continuous in ξ and Q. Per definition, for any k, i, c, $\mathbb{E}^Y_{k,c}[\delta U^i_{k,c}] = 0$ and $\{\sum_{j=0}^k \delta U^i_{k,c}\}_k$ is a martingale sequence. We have therefore shown that the iterates eq. (5) can be written as the multi-scale stochastic approximation iterates of theorem 4.

If the noise sequences $(\xi^i_{k,c})$ satisfy **A** A.2 to A.4, then according to theorem 4 the iterates follow the $M \times C$ mean ODEs:

$$\frac{d}{dt} q^i_t(s^i_c, a^i_c) = \frac{1}{u^i_c} \bar{F}^i(q^{<i}_t, q^i_t, Z^{\geq i+1}(q^{<i+1}_t)) \tag{20}$$

where $\pi^{-i} = (\phi(q^1_t), \dots, \phi(q^{i-1}_t), \phi(q^{i+1}_t)), \dots, \phi(q^M_t))$, the $\{q^j_t\}_{j<i}$ are constant, $\{q^j_t\}_{j>i} = Z^{>i+1}(q^{<i+1}_t)$. Then, assumption **A** A.9 guarantees that (20) admits an asymptotically stable point, and we conclude on the convergence of the iterates towards a smooth equilibrium as defined in definition 2.

We now need to prove that the noise sequences $\xi^i_{k,c}$ with values in the space Ξ defined in eq. (19) are Markovian state-dependent noise sequences, satisfying **A** A.2 and **A** A.4. Let us derive an expression for $\mathbb{E}^Y_k[Y^i_{k,c}]$. First, $\hat{Q}^i_{k,c}$ is a function of $\hat{Q}^i_{0,c}$ and the previous $Y^i_j, \tau^i_j, j < q$, so we only need to focus on s^i_{k+1} and r_k. The next state s^i_{k+1} is sampled from the local transition kernel after the agent has visited component (s^i, a^i), so we have exactly:

$$\mathbb{E}^Y_{k,c}\left[[\phi(\hat{Q}^i_k)(s^i_{k+1})]^T \hat{Q}_k(s^i_{k+1}, \cdot)\right] = v'(\hat{Q}^i_k, (s^i, a^i)_c)$$

As for the reward r_k: $\mathbb{E}^Y_{k,c} r_k = \mathbb{E}^Y_{k,c} r(s_k, a_k) = \mathbb{E}^Y_{k,c} r((s^i, a^i)_c, s^{-i}_k, a^{-i}_k)$. Neither s^{-i}_k nor a^{-i}_k are observed by agent i. If s^{-i}_k was known however, then the expectation of a^{-i}_k would just be taken from the respective policies of other agents at that time $\bar{\pi}^{-i} = \phi(\hat{Q}^{-i}_k)$:

$$\mathbb{E}^Y_{k,c} r_k = \mathbb{E}^Y_{k,c}\left[\mathbb{E}^Y_{k,c}\left[r((s^i, a^i)_c, s^{-i}_k, a^{-i}_k) \mid s^{-i}_k\right]\right] = \mathbb{E}^Y_{k,c}\left[[\phi(\hat{Q}^{-i}_k)(\cdot, s^{-i}_k)]^T r((s^i, a^i)_c, s^{-i}_k, \cdot)\right]$$

It remains to handle s^{-i}_k. It is easy to see that the state process $\{s^{-i}_k\}_k$ is in fact a Markovian process, whose transition kernel depends on the iterates $\hat{Q}_k$ in real time. Recall that P is the global transition matrix of dimension $|S| \times |A_i| \dots |A_M| \times |S|$. By construction the mapping ϕ returns a policy assigning a non-zero probability to every action, so that there exists $\epsilon_\phi > 0$ such that for all $a \in A_i$, $\pi(a|s^i) > \epsilon_\phi$. For an initial distribution d_0, we write $\{d_k\}_{k\geq 0} \in (0,1)^{|S|}$ the process tracking the distribution of s^{-i}:

$$d_{k+1} = d_k \cdot P\Pi^M_{j=1}\phi(\hat{Q}^j_k)$$

$\{d_k\}_k$ is a state-dependent Markovian process, that is:

$$P(d_{k+1} \in \cdot | \mathcal{F}^{i,Y}_{k,c}, d_k) = P(d_{k+1} \in \cdot | \hat{Q}_k, d_k) \tag{21}$$

We can now write the processes $\{\xi^i_{k,c}\}_k$ as:

$$\xi^i_{k,c} = \sum_s [d_k(s|s_i) - d^{\pi_k}(s|s_i)]\left[\phi(Q^{-i}_k)(\cdot, s^{-i}_k)]^T r(s^i_c, s^{-i}_k, a^i_c, \cdot)\right] \tag{22}$$

where $d^{\pi_k} = d^{\phi(Q_k)}$ is still the stationary distribution over global states under policy $\phi(Q_k)$ as defined by:

$$d^\pi(s|s^i) = P(\hat{\mathbf{s}} = s \mid \hat{\mathbf{s}}^i = s^i) = \frac{P(\{\hat{\mathbf{s}} = s\} \cap \{\hat{\mathbf{s}}^i = s^i)\}}{P(\hat{\mathbf{s}}^i = s^i)} = \frac{1_{[\hat{\mathbf{s}}(i) = s^i]} d^\pi(s)}{\sum_{\bar{s}} 1_{[\bar{s}(i) = s^i]} d^\pi(\bar{s})} \tag{23}$$

. Since the reward is bounded in $[-R, R]$, the $\{\xi^i_{k,c}\}_k$ take values in the compact $[-R|S||A|, R|S||A|]$. The $\{\xi^i_{k,c}\}_k$ being an affine transformation of $\{d_k\}$, it follows that it is also a state-dependent Markovian process. Moreover, this state-dependent process is stationary, in the sense that for each Q there is a time-invariant (does not depend on k if we know Q) measurable transition function $P^\xi(\cdot, \cdot|Q)$ such that $P(\xi_{k+1,c} \in \cdot | \mathcal{F}^{i,Y}_{k,c}, d_k) = P^\xi(\xi_{k,c}, \cdot|\hat{Q}_k)$. Therefore, **A** A.2 is satisfied.

It remains to show that the noise $\{\xi_i\}_k$ satisfies **A** A.4: its "rate of change" is small enough that it can be locally averaged out, and the noisy observations can be approximated by the mean ODE. In particular, we define the fixed $Q-$chain $\{\xi_{k,c}(Q)\}$, the Markov chain on state space Ξ with the fixed transition function $P(\cdot, \cdot|Q)$. It is the noise process starting from n if $\hat{Q}$ stayed constant forever: $\{\xi_{n+j,c}(\hat{Q}), j \geq 0, \xi_{n,c}(\hat{Q}) = \xi_{n,c})\}$. To verify **A** A.4, we need to prove for any compact set $A \in \Xi$,

$$\lim_{n,m} \frac{1}{m} \sum_{l=n}^{n+m-1} \mathbb{E}^Y_n\left[\xi_{l,c}(\hat{Q}) \mathbb{I}_{\{\xi_{l,c} \in A\}}\right] = 0 \tag{24}$$

We define the corresponding fixed $Q-$chain $\tilde{d}_{n+j}(s|s_i, Q)$, for all $j \geq 0$ such that:

$$\tilde{d}_n = d_n, \quad \tilde{d}_{k+1} = \tilde{d}_k \cdot P\Pi^M_{j=1}\phi(Q^j) = \tilde{d}_k \cdot P^Q \tag{25}$$

Switching to vector notation, we write R_π the vector of size $|S|$ of reward expectations under the global policy π for the global state s. So for all $s \in S$,

$$R_{\pi_k}(s) = \left[\phi(Q^{-i}_k)(\cdot, s^{-i}_k)]^T r((s^i, a^i)_c s^{-i}_k, \cdot)\right]$$

We also write $\tilde{D}_l(Q)$ and $D(Q)$ the corresponding state distribution vectors for $\tilde{d}_l(s|s_i, Q)$ the fixed Q-chain starting in n as defined in eq. (25) and $d^{\phi(Q)}(s|s_i)$ the stationary distribution under policy $\phi(Q)$.

Then for all n, m, and any $\hat{Q}$ putting (22) into (24) allows us to rewrite the latter as:

$$\begin{aligned}
&\frac{1}{m}\sum_{l=n}^{n+m-1}\mathbb{E}_n^Y\left[\xi_{l,c}(\hat{Q})\mathbb{I}_{\{\xi_n\in A\}}\right] = \frac{1}{m}\sum_{l=n}^{n+m-1}\left[\left(\tilde{D}_l(\hat{Q})-D(\hat{Q})\right)R_{\pi_l}\right]\mathbb{I}_{\{\xi_l\in A\}}\\
&= \frac{1}{m}\sum_{l=n}^{n+m-1}\left[\left((\tilde{D}_n(\hat{Q})\Pi_{j=n}^{l-1}P^{\hat{Q}})-D(\hat{Q})\right)R_{\pi_l}\right]\mathbb{I}_{\{\xi_l\in A\}}\\
&= \frac{1}{m}\sum_{l=n}^{n+m-1}\left[\left(d_n\left(P^{\hat{Q}}\right)^{l-n}-D(\hat{Q})\right)R_{\pi_l}\right]\mathbb{I}_{\{\xi_l\in A\}}\\
&\leq \frac{R}{m}\sum_{l'=0}^{m-1}\left\|\left(d_n\left(P^{\hat{Q}}\right)^{l'}-D(\hat{Q})\right)\right\|_1\mathbb{I}_{\{\xi_{l'+n}\in A\}}
\end{aligned} \tag{26}$$

From **A** 2.2, we know that the finite Markov chain representing the global state process is irreducible and aperiodic. Therefore, $P^{\hat{Q}}$ is the transition matrix associated with an irreducible global state process over the finite state-space S, and the stationary distribution defined by $D^{\phi(\hat{Q})}$ is its limiting state distribution. Moreover, the convergence rate is geometric [21], so that for any initial distribution d_n there exists constants $0<b<1$ and $C>0$ such that for all l:

$$\left\|\left(d_n\left(P^{\hat{Q}}\right)^{l}-D(\hat{Q})\right)\right\|_1 < C(1-b)^l$$

Therefore, together with (26) we have that:

$$\lim_m\lim_n\frac{1}{m}\sum_{l=n}^{n+m-1}\mathbb{E}_l^Y\left[\xi_{l,c}(\hat{Q})\mathbb{I}_{\{\xi_n\in A\}}\right]\leq\lim_m\frac{R}{m}\sum_{l'=0}^{m-1}C(1-b)^{l'}=\lim_m\frac{CR}{mb}=0$$

□

A.5 Convergence with acyclic dependence structure

Proof. We recall the mean ODE followed by each agent i as introduced in (20):

$$\frac{d}{dt}q_t^i(s_c^i,a_c^i)=\frac{1}{u_c^i}\bar{F}^i(q_t^{<i},q_t^i,Z^{\geq i+1}(q_t^{<i+1}))$$

with $\bar{F}^i(q_t^{<i},q_t^i,Z^{\geq i+1}(q_t^{<i+1}))=r(s_c^i,a_c^i,q_t^{<i},q_t^i,Z^{\geq i+1}(q_t^{<i+1}))+\beta\sum_{s'}P(s^i,a^i,s'^i)q^i(s'^i,\phi(q)(s'^i))$ and $r_{q^i}(s_c^i,a_c^i,q_t^{<i},q_t^i,Z^{\geq i+1}(q_t^{<i+1}))=\sum_s d^{\pi_{q^i},Z^{>i+1}(q^i)}(s)\sum_{a^{-i}}\pi^{-i}_{q^i,Z^{>i+1}(q^i)}(a^{-i})r(s,a^i,s^{-i})$. According to **A** 3.1 and for each agent i and component c, the mapping from q^i to $r_{q^i}(s_c^i,a_c^i,q^i)$ is a K - contraction mapping. $\bar{F}^i$ is therefore a $(K+\beta)$ contraction mapping with regard to the infinite norm. It follows that for each agent i there is a unique fixed point Q^{*i} such that $\bar{F}^i(Q^{<i},Q^{*i},Z^{\geq i+1}(Q^{<i+1}))=Q^{*i}$ and that this fixed point is the unique globally asymptotically stable point of the ODE $\dot{X}=\bar{F}^i(Q^{<i+1},Z^{\geq i+1}(Q^{<i+1}))$. Recall that the reflection terms are null. The multiplication by the factor $1/u_c^i$ has a time scaling effect on the ODE but does not change its asymptotic behavior. It follows that **A** A.9 on the asymptotic behaviors of the mean ODEs is satisfied. A sequence of learning rates α_k^i has been assigned to each agent i such that **A** A.5 and A.8 are satisfied. The weak convergence of the iterates $\hat{Q}_k$ towards a smoothed equilibrium then follows from theorem 5. □

A.6 Learning rates attribution procedure

We call `TopSort` any topological sorting algorithm.

We call `RBFS` the procedure that does reverse breadth first search on the graph, starting from a single leaf up to all discovered roots, and returns the set of all visited nodes.

The detail of this algorithm is quite straightforward. First, Line 3 a total order on all nodes is extracted through a topological sort. As noted earlier this is already a valid ranking for any DAG structure that will ensure convergence, and the rest of the algorithm is dedicated to reduce the number of ranks if possible.

Lines 4 to 9 simply extract constraint sets from the DAG: two nodes belonging to the same set admit at least a path towards a same third node, and therefore cannot receive the same rank. This is done by returning, for every leaf, the set of all its ancestors through the reverse breadth first search.

Then, lines 11 to 21 simply adjust the ranking by extracting node by topological order, and assigning them the slowest rank that does not conflict with other nodes in his constraint sets. The final ranking, by construction, ensures that all nodes have higher ranks than their parents (through the topological order) and different ranks from other nodes in their constraint sets.

Algorithm 1 Procedure: Attribute learning rates for any DAG structure

Require: Network DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with M nodes
Init $\Sigma \leftarrow \{\}, d \leftarrow [\infty, \ldots, \infty] \in (\mathbb{N} \cup \infty)^M$, Heap Q
$\mathcal{V}_R \leftarrow$ Leaves($\mathcal{G}$)
$Q \leftarrow TopSort(\mathcal{G})$
Retrieve constraint sets Σ
for leaf in $\mathcal{V}_R$ **do**
  $A \leftarrow RBFS$(leaf)
  $\sigma \leftarrow A \bigcup \{\text{leaf}\}$
  Σ.append(σ)
end for
Assign levels based on Σ
while Q not empty **do**
  $v \leftarrow Q.pop()$
  if $Parents(v)$ empty **then**
    $d^v \leftarrow 0$
  else
    $d^v \leftarrow \max(\{d^p : \quad p \in Parents(v)) + 1$
  end if
  $\bar{C} \leftarrow \{d^n : \exists \sigma \in \Sigma, \sigma \supset \{n, p\}\}$
  while $d^v \in \bar{C}$ **do**
    $d^v \leftarrow d^v + 1$
  end while
end while
return d

A.7 Proof of theorem 2

We detail the proof of theorem 2. The case handled here is slightly more general that the one in the theorem. We start by considering a set of learning rate sequence attribution that preserves the convergence of theorem 5. We then show that `TreeLRs` returns a learning rate distributions that belongs to that set.

Proof. We note that for a node i, a best response $\pi*^i$ to other policies π^{-i} only depends on policies in the neighborhood. Any change at a local policy outside of the neighborhood does not lead to a change in best-response to π^{-i} if local policies in the neighborhood remain the same.

That is, let i be a node in the direct acyclic graph $\mathcal{G}$, and π^i the local policy followed by agent i. We remind that $\mathcal{N}^i$ is the set of the neighbors of i and write the corresponding set of local policies $\pi^{\mathcal{N}^i} = \{\pi^j, j \in \mathcal{N}^i\}$. Similarly, we write the set of the remaining nodes excluding $i, \bar{\mathcal{N}}^i = \mathcal{V} \backslash \mathcal{N}^i$ and their local policies $\pi^{\bar{\mathcal{N}}^i}$. Let local policy π^{*i} be a best response to $\pi^{-i} = (\pi^{\mathcal{N}^i}, \pi^{\bar{\mathcal{N}}^i})$. Then, for any other set of local policies $\pi'^{\bar{\mathcal{N}}^i}$, π^{*i} is also a best response to $\pi'^{-i} = (\pi^{\mathcal{N}^i}, \pi'^{\bar{\mathcal{N}}^i})$. This is immediate if we recall that for π^{*i} is a best response to π if for any local policy π^i we have for all $s^i \in S_i$ and $a^i \in A_i$, $Q^{\pi^{-i}}_{\pi*^i}(s^i, a^i) \geq Q^{\pi^{-i}}_{\pi^i}(s^i, a^i)$, and that due to the structure of the reward as a sum of local components:

$$Q^{\pi^{-i}}_{\pi^i}(s^i, a^i) = \sum_{l=1}^{M} \mathbb{E}_{\pi^l, \pi^{U^l})} G(l, U^l)$$

with

$$\mathbb{E}_{(\pi^l, \pi^{U^l})} G(l, U^l) = \mathbb{E}_{(s^l_0, s^{U^l}_0, a^l_k, a^{U^l}_k, s_k) \sim (d^{\pi^l}, d^{\pi^{U^l}}, \pi^l, \pi^{U^l}, P)} \left[\sum_{k=0}^{\infty} \beta^k r^l(s^l_k, a^l_k, s^{U^l}_k, a^{U^l}_k) \mid s^i_0 = s^i, a^i_0 = a^i \right]$$

and

$$Q^{\pi^{-i}}_{\pi^i}(s^i, a^i) = \sum_{l \in (i, \mathcal{N}^i)} \mathbb{E}_{(\pi^l, \pi^{U^l})} G(l, U^l) + \sum_{l \in \bar{\mathcal{N}}^i} \mathbb{E}_{(\pi^l, \pi^{U^l})} G(l, U^l)$$

Now let local policy π^{*i} be a best response to $\pi^{\mathcal{N}^i}$ only, and take any set of local policies excluding i $\pi^{-i} = (\pi^{\mathcal{N}^i}, \pi^{\bar{\mathcal{N}}^i})$. Then $\sum_{l \in (i, \mathcal{N}^i)} \mathbb{E}_{\pi^l, \pi^{U^l})} G(l, U^l) \geq \sum_{l \in (i, \mathcal{N}^i)} \mathbb{E}_{\pi^l, \pi^{U^l})} G(l, U^l)$ per definition of the

best response. The second sum is a constant with respect to the local policy of i, and we have $Q_{\pi^{*i}}^{\pi^{-i}}(s^i, a^i) \geq Q_{\pi^i}^{\pi^{-i}}(s^i, a^i)$.

For now, we assume that learning rates are such that there exists no two paths from two different agents of the same level towards the same node i, and show that the result will follow. Note that this indeed explicitly the property we ensure if we follow the procedure to assign learning rates (see. appendix A.6) for the case of a common DAG.

At each i, iterates for agents i will follow an ODE whose mean field will be a continuous function of both the parameters of agents in $\mathcal{N}_{in}^i$, which by construction learn on a slower scale, and the parameters of agents in $\mathcal{N}_{out}^i$, which by construction learn on a faster scale. At the scale of i, the parameter of $\mathcal{N}_{in}^i$ can be considered fixed. The parameters of $\mathcal{N}_{out}^i$ will have converged towards the parents of $\mathcal{N}_{out}^i$. If the $\mathcal{G}$ is a tree, then the set of parents of $\mathcal{N}_{out}^i$ is reduced to $\mathcal{N}_{in}^i$, and convergence follows. If $\mathcal{G}$ is any standard DAG, then the mean field of any agent $j \in \mathcal{N}_{out}^i$ is $\propto \bar{F}^j(q^{U^j})$. Any node $l \in U^j$ is either slower than i, in which case it can be considered constant, or faster than i, in which case it has converged towards a continuous function of parameters q^{U^l}. Again, any node k in U^l is either slower than i, or has converged towards a continuous function of parameters q^{U^k}. By repeating this procedure, we find that $\bar{F}^j(q^{U^j})$ depends on all the ancestors of j. Recall that the graph is acyclic, and we have just assumed that there never exists two paths from two agents at the same level towards any node. Therefore we can rewrite $\bar{F}^j(q^{U^j})$ as a function of a finite number of agents, in which any node that has the same level with i must be i itself. Therefore, at the timescale i, j will have converged towards a continuous function of q^i and a set of paramters slower than i. Convergence of i follows.

Now it is easy to show that when $\mathcal{G}$ is a tree, and we follow the procedure `TreeLRs`, then there never exists two paths from two agents of the same level towards any node i. Indeed, if there existed such two nodes v and $\bar{v}$ with a path towards i, then in a tree there would also exist a path between v and $\bar{v}$, and the level of v (or $\bar{v}$) would be strictly greater than the level of $\bar{v}$ (or v). This concludes the proof. □