Multi-agent reinforcement learning for partially observable cooperative systems with acyclic dependence structure

Claire Bizon Monroc, Ana Bušić, Donatien Dubuc, Jiamin Zhu

Multi-agent reinforcement learning for partially observable cooperative systems with acyclic dependence structure

Claire Bizon Monroc *, Ana Bušić †, Donatien Dubuc , and Jiamin Zhu ‡

Abstract. Single-agent reinforcement learning algorithms can be directly applied to multiagent systems in an independent learning approach, but they then lose any convergence properties due to non-stationarity. We prove that in transition-independent Decentralized Partially Observable Decentralized Markov Decision Process (Dec-POMDP) non-stationarity can be mitigated by a multi-scale approach when the interdependence of agents dynamics can be represented by a directed acyclic graph (DAG). We propose a multi-scale Q-learning algorithm (MQL) where agents update local q-learning iterates at different timescales without communication and still converge. To this purpose, we first show that we can model the loss of information on the global state as a state-dependent Markovian noise. Then, we show that results from stochastic approximation theory can be used to prove the convergence of the MQL under partial state observability. Next, we give practical solutions to exploit knowledge about agent interaction to assign learning rates that ensure convergence, and propose a NetworkMQL algorithm that can achieve convergence in Network-Distributed POMDP (ND-POMDP). Finally, we validate both MQL and NetworkMQL on a wind farm control problem from the energy industry.

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 [36, 22]. Multi-agent reinforcement learning (MARL) has encountered successes in fields as varied as games with multiple players [3], vehicle routing problem for traffic regulation [35], or distributed optimal control of wind farms [28]. In this article, we 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 processe (Dec-POMDPs) in the MARL literature. It is known that solving Dec-POMDPs is very hard: finding a solution to a two-player Dec-POMPD has been proven to be NEXP-hard, and is undecidable for the infinite-horizon case [6]. Instead of general Dec-POMDP, we focus on the special case of the *transition-independent* Dec-POMDP, which is of NP-hard complexity [1]. 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 [5] 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.

Note that transition-independent Dec-POMDPs can also be constructed from certain standard Dec-POMDPs, and that such a reformulation can be useful to solve real industrial prob-

*C. Bizon Monroc is with Inria and DI ENS, École Normale Supérieure, PSL Research University, Paris, France and IFP Energies nouvelles. E-mail: claire.bizon-monroc@inria.fr

†A. Bušić is with Inria and DI ENS, École Normale Supérieure, PSL Research University, Paris, France.

‡D. Dubuc and J. Zhu are with IFP Energies nouvelles.

lems. In [18] for example, a distributed wind farm optimization problem is considered, in which the local information of the Dec-POMDP can be factorized into two components. The first is a private component, that is a local component independent of other agents. The second is a deterministic function of the private components of other agents, and of an exogenous markovian process that is independent of any agent's action. Whenever such an exogenous process can be identified, 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. A detailed example will be given in Section 5.

Transition-independent Dec-POMDPs have been studied in the planning literature, and several planning algorithms have been proposed to find optimal local policies [21, 5, 12, 10]. However, they exploit the full knowledge of the transition matrix and reward function, which are rarely available in real-life problems. MARL algorithms, on the other hand, focus on learning solely from interactions with the environment. If a single learner updates all local policies based on the global observation, one says that the training is centralized. Multi-agent systems are however often under constraints that prevent instantaneous communication of all agents with a central controller. Moreover, centralized approaches are vulnerable to the curse of dimensionality, as the size of the search space increases exponentially with the number of agents.

In an online decentralized training approach, every agent rather only updates its own local policy with the information it has collected. To compensate for the loss of information, online learning algorithms for Dec-POMDP typically allow agents to keep past observations in their memory, allow communication between agents during training, maintain beliefs about the global state of the environment, or estimate the behaviors of other agents [21, 22]. Instead, an interesting question is to see whether memory-less policies can still be learned without additional estimates or communication. In particular, MARL has focused on extending the success of single-agent reinforcement learning algorithms to the multi-agent case, like the classic Q-learning algorithm [33]. One naive way to adapt the single-agent Q-learning to the multi-agent case is to simply let each agent run local Q-learning updates, with other agents considered as a part of its environment. This approach was first introduced by [31] as *Independent learning*. Independent learning has been shown to produce good empirical results in a number of problems [17], and is used as a baseline to develop current state-of-the-art MARL algorithms [34]. Because it avoids the exponential dependence of dimension of the search space in the number of agents, independent learning is more scalable. It has however no convergence guarantee: agents acting - and learning - simultaneously make the state and reward processes appear non-stationary for each agent, an issue that is commonly referred to as the *non-stationarity* problem. Moreover, the system may prevent full observability of the environment, with each agent only collecting a function of the global state.

Solutions to mitigate the non-stationarity problem exist, and typically rely on a modification of algorithm's local rule. Examples have been turn-based control [25] or turn-based updates [29], two stages algorithms requiring agent coordination during learning [27, 2], coordinated exploration between agents, and consensus-based approaches or explicit modeling of other agent's behavior [36].

In this article, we show that for independent learners in transition-independent Dec-

POMDP, the loss of information due to partial observability can be seen as a Markovian noise. We then focus on the case where agent dynamics can be described by a directed acyclic graph (DAG), and show that in such a case the non stationarity issue can be addressed by a multi-timescale learning approach: by allowing agents to learn at different time scale, we ensure that for any "fast" agent, the "slow" evolution of other policies will lead iterates to behave as though the environment was stationary, and ultimately converge. Note that this approach belongs to the independent learning family and thus is scalable to large number of agents.

Our convergence analysis will rely on stochastic approximation techniques. Indeed, stochastic approximation and multi-timescale approaches have been used to analyze fully observable learning algorithms. In [8], the convergence of the single-agent Q-learning has been proven with a stochastic approximation approach. Two-timescale stochastic approximation was then introduced in [7], showing that two interdependent stochastic processes can both converge when they are updated at two different scales. 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 [4, 26, 16] and Q-learning [24] inspired schemes for fully observable zero-sum and team games. Two-timescale results have been extended to an arbitrary number of scales in [15], and used to analyze a multi-scale learning algorithm in some classes of repeated games. In [20], a similar multi-scale approach is further evaluated on several multi-agent reinforcement learning problems, but an analysis of its convergence in this case is not provided. Unfortunately, these convergence results have required 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.

We exploit weak convergence results from the stochastic approximation theory including state-dependent noise [14]. We first formally extend the weak convergence [Theorem 6.2, Chapter 8, [14]] for synchronous two-timescale iterates to synchronous multi-timescale iterates (see Theorem 3.1), then further extend this result together with [Theorem 5.1, Chapter 12, [14]] for single-timescale asynchronous updates to asynchronous multi-timescale updates (see Theorem 3.2). Then, using these results, we prove that our multi-scale Q-learning converges under carefully chosen learning rates (see Theorem 3.3). More precisely, we show that our multi-scale Q-learning algorithm can be framed as multi-scale stochastic approximation updates with state-dependent noise, where the tracking error of the global state can be modeled as a latent Markovian process and satisfies necessary assumptions for applying Theorem 3.2. Next, we propose a faster algorithm and establish convergence result (see Theorem 4.2) for a Dec-POMDP with acyclic dependence structure between agent dynamics. In particular, we build on the network distributed POMDP problems [19], in which interactions between agents can be represented by a sparse graph. We 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 propose a multi-scale Q-learnning algorithm. In Section 3, we establish weak convergence for multi-timescale iterates, and then apply it to analyze and prove the convergence of our multi-scale Q-learning. Then, in Section 4, 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. Our experiment in Section 5 then evaluates the multi-scale approach on the real industrial problem of wind farm control, and empirically validates its convergence.

2. Cooperative MARL with local learners.

We start by formalizing 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 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, the global action space A is factorized into M local action spaces $A = A$. 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 $\{\pi^1, \ldots, \pi^M\}$ and we write $\pi = (\pi^1, \ldots, \pi^M)$. 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 [11] 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 [5]: every agent's local state is only influenced by its own current state and action.

A 2.1. *We assume that transitions between locally observed states only depend on local state and actions. That is, there are local transition kernels $P^i_{i=1\ldots M}$ 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.

A 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 [33, 32, 13] 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.

Define $d^\pi(\cdot|s^i)$ the conditional distribution over global states in S conditional on observing s^i. For any $i \in \{1,\dots,M\}$, $\hat{\mathbf{s}} \sim d^\pi$, $\hat{\mathbf{s}}^i \sim d^{\pi^i}$, and $s, s^i \in S \times S_i$, we have

$$d^\pi(s|s^i) = P(\hat{\mathbf{s}} = s \mid \hat{\mathbf{s}}^i = s^i) = \frac{P(\{\hat{\mathbf{s}} = s\} \cap \{\hat{\mathbf{s}}^i = s^i)\}}{P(\hat{\mathbf{s}}^i = s^i)} = \frac{1_{[\hat{\mathbf{s}}(i)=s^i]} d^\pi(s)}{\sum_{\bar{s}} 1_{[\bar{s}(i)=s^i]} d^\pi(\bar{s})} \tag{2.1}$$

Since the local state is a deterministic function of the global state, this conditional distribution depends only on the marginal stationary distribution of the global state.

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

2.2. Multi-scale Q-learning. 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:

$$Q_{\pi^i}^{\pi^{-i}}(s^i,a^i) = \mathbb{E}_{s_0 \sim d^\pi, a_k \sim (\pi^i,\pi^{-i}), s_k \sim P}\left[\sum_{k=0}^{\infty} \beta^k r(s_k,a_k) \mid s_0^i = s^i, a_0^i = a^i\right] \tag{2.2}$$

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 2.1.

Lemma 2.1. *Any local q-function (2.2) satisfies the following recursive formula:*

$$\begin{aligned} &Q_{\pi^i}^{\pi^{-i}}(s^i,a^i) \\ &= \sum_s d^\pi(s \mid s^i) \sum_{a^{-i}} \pi^{-i}(a^{-i}|s) r(s,a) + \beta \sum_{s'^i} P^i(s^i,a^i,s'^i) \sum_{a'^i} \pi^i(a'^i|s'^i) Q_{\pi^i}^{\pi^{-i}}(s'^i,a'^i) \end{aligned} \tag{2.3}$$

The proof of Lemma 2.1 is straightforward but tedious, we detail it in Appendix A. Like for the single-agent q-value function [30], 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 (2.2), 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 2.2 (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 $\pi^i(\cdot \mid s^i) \in \arg\max_{\rho \in \Omega(s^i)} \left[\rho^T Q^{\pi^{-i}}_{\pi^i}(s^i,\cdot)\right]$, 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 [15, 24]: for any given q-value table Q^i, let us define the mapping ϕ that returns the following local policy:

$$\phi(Q^i)(\cdot \mid s^i) = \arg\max_{\rho \in \Omega(s^i)} \left[\rho^T Q^i(s^i,\cdot) + \tau \nu^i_{s^i}(\rho)\right] \quad \forall s^i \in S_i \tag{2.4}$$

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 2.3 (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:

$$v'(Q^i,s^i,a^i) = \sum_{s'^i} P^i(s^i,a^i,s'^i)[\phi(Q^i)(\cdot \mid s'^i)]^T Q^i(s'^i,\cdot) \tag{2.5}$$

Then writing $Q^{\pi^{-i}}_* = Q^{\pi^{-i}}_{\pi^{*i}}$, from Lemma 2.1 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 $i \in \{1 \dots M\}$, $s^i \in S_i$, $a^i \in A_i$. Let all agents maintain a local estimate $\hat{Q}^i$ of the q-function (2.2), and follow a local policy $\pi^i = \phi(\hat{Q}^i)$. The combined actions of all agents sample M local trajectories $\{(s_0^i, a_0^i, r_0^i), (s_1^i, a_1^i, r_1^i) \dots\}, i \in \{1 \dots M\}$. Let now all agents locally run a Q-learning update, so that each agent updates its local estimate $\hat{Q}_k^i$ at each timestep k:

$$\begin{aligned} \hat{Q}_{k+1}^i(s^i, a^i) = \hat{Q}_k^i(s^i, a^i) \\ + \alpha_k^i(s_k^i, a_k^i) \left[r_k + \beta[\phi(\hat{Q}_k)(s_{k+1}^i)]^T \hat{Q}_k(s_{k+1}^i, \cdot) - \hat{Q}_k^i(s_k^i, a_k^i)) \right] I_{k,s^i,a^i} \end{aligned} \tag{2.6}$$

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.

We will show that these iterates can converge when learning rates are carefully chosen, and call the resulting algorithm the *multi-scale Q-learning* algorithm. 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 (2.2). In fact, no agent ever collects a reward sampled from any stationary distribution at all. Instead, we will 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 will exploit results from the stochastic approximation theory concerning multi time scales iterates with Markovian noise.

In the next section, we will first establish the weak convergence of a general multi-scale algorithm in the synchronous and asynchronous cases. Then, these convergence results will be used to prove the convergence of iterates (2.6) in Subsection 3.3.

3. Weak convergence of the multi-scale algorithms.

3.1. Weak convergence of synchronous multi-scale iterates with Markovian noise.

Weak convergence of stochastic approximation for two time-scales systems were proven in [14]. 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 $i \in \{1, \dots, d\}$ 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 $\theta_k^1, \dots, \theta_k^M$ updated according to iterates:

$$\theta_{k+1}^i = \Pi_H \left[\theta_k^i + \alpha_k^i Y_k^i \right] = \theta_k^i + \alpha_k^i (F^i(\theta_k, \xi_k^i) + \delta U_k^i) + B_k^i \tag{3.1}$$

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

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

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:

A 3.1. *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** 3.5), 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.

A 3.2. *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 proof of Lemma B.4) 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:

A 3.3. *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 $P^i(\xi, \cdot|\theta_j)$.*

A 3.4 (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:

$$\theta_{\alpha^j}^{i,k}(t) = \theta_{\alpha^j}^{i,0}(t_k^j + t) \tag{3.2}$$

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.

A 3.5 (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.

A 3.6. *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

$$Z^{\geq i}(\theta_k^{<i}) = (\zeta^i(\theta_k^{<i}), Z^{\geq i+1}(\theta_k^{<i}, \zeta^i(\theta_k^{<i}))), \quad i = 2, \cdots, M-2 \tag{3.3}$$

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 4, we will introduce specific DAG structures on agent interaction that can satisfy them, and a concrete example will be given in Section 5.

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 (3.1) in the following theorem.

Theorem 3.1 (Weak convergence of multi-scale iterates with Markovian noise). *Consider iterates (3.1). Let $\{\theta_{\alpha^j}^{i,k}(\cdot)\}$ be the interpolation of the process θ_k^i on the j-th timescale, defined by (3.2). If **A** 3.1-3.5 hold, then $\{\theta_{\alpha^1}^{1,k}(\cdot)\}$ admits a subsequence which converges towards a process $\theta^1(\cdot)$ such that:*

$$\dot{\theta^1} = \bar{F}^1(\theta^1, Z^{\geq 2}(\theta^1)) + b^1, \quad b^1(t) \in -\Upsilon(\theta^1(t)) \tag{3.4}$$

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 (3.4) on the interval $[0, T]$ goes to one in probability as $T \to \infty$.

The proof of this theorem is detailed in Appendix B. Note that Theorem 3.1 is an extension of the week convergence result established for two timescale iterates by [Theorem 8.6.1, [14], p.286], and the extension procedure to multi-scale is inspired by [15]. The idea behind Kushner's original proof in [14] 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.

3.2. 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 $\alpha^i_k = \alpha^i_{k,1} = \alpha^i_{k,2} = \cdots = \alpha^i_{k,C}$. The M iterates in (3.1) can therefore be seen as $M \times C$ iterates, with the updates to each component following:

$$\theta^i_{k+1,c} = \Pi_H \left[\theta^i_{k,c} + \alpha^i_{k,c} Y^i_{k,c}\right] \tag{3.5}$$

The time between the kth and $(k+1)$th updates of the element indexed by c in $\{\theta^i_k\}_k$ is given by the random variable $\tau^i_{k,c}$. 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 $\{\theta^i_k\}_k$ is done at the real time $T^i_{k,c} = \sum_{n=0}^{k-1} \tau^i_{n,c}$. We note $\Gamma^i_{k,c} = \sum_{n=0}^{k-1} \alpha^i_n \tau^i_{n,c}$ the corresponding scaled real time, and introduce the real-time interpolation $\hat{\theta}^i_c$: $\hat{\theta}^i_c(t) = \theta^i_{k,c}$ on $[T^i_{k,c}, T^i_{k+1,c})$. Like in (3.2), we look at the shifted piecewise constant interpolations θ^i_{c,α^j} of the sequences $\{\theta^i_{k,c}\}_k$ at every timestep $j = \{1, \ldots, C\}$ 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

$$\theta^{i,\upsilon}_{c,\alpha^j}(t) = \theta^i_{k+p^i_c,c}, \quad t \in [t^{ij,\upsilon}_{k,c}, t^{ij,\upsilon}_{k+1,c}), \qquad t^{ij,\upsilon}_{k,c} = \sum_{n=p^i_c(\upsilon)}^{k-1} \alpha^j_n \tag{3.6}$$

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

$$\hat{\theta}^{i,\upsilon}_{c,\alpha^j}(t) = \theta^i_{k,c}, \quad t \in [\Gamma^{ij,\upsilon}_{k+p^i_c,c}, \Gamma^{ij,\upsilon}_{k+1,c}) \qquad \Gamma^{ij,\upsilon}_{k,c} = \sum_{n=p^i_c(\upsilon)}^{k-1} \alpha^j_n \tau^i_{n,c} \tag{3.7}$$

We now extend the definitions of the $\sigma-$algebra used in Subsection 3.1. 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:

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

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** 3.1-3.5 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:

A 3.7. *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.*

A 3.8. *Every component's learning rate $\alpha^i_{k,c}$ can be written as a local average of positive real-valued functions f^i:*

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

A 3.9. *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 $i \geq 2$:*

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

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 (3.3)

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

Theorem 3.2 (Weak convergence of asynchronous multi-scale iterates with Markovian noise). *Consider iterates (3.5), updated asynchronously following the time interval sequences $\{\tau^i_{k,c}\}_k$. If Assumptions **A** 3.1-3.5 hold, and Assumptions **A** 3.7-3.9 also hold, then the conclusion of Theorem 3.1 still holds with the limit process:*

$$\dot{\hat{\theta}}^1_{c,\alpha^1}(t) = \frac{\bar{F}^1(\theta^1_{c,\alpha^1}(t), Z^{\geq 2}(\theta^1_{c,\alpha^1})(t))}{u^i_c(t)} + \hat{b}^1_{c,\alpha^1}(t) \qquad u^i_c(t) \in [1, \bar{u}^i_c]. \tag{3.8}$$

The proof of this theorem is laid out in Appendix C. Note that the weak convergence of asynchronous updates for the single-agent case has been established in [Theorem 12.3.5 [14]], and we extend it to the multi-scale case. As previously in the proof of Theorem 3.1, we derive 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 is then 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 (3.9).

3.3. Weak-convergence of multi-scale Q-learning iterates. Let us now look at multi-scale Q-learning iterates (2.6). Agents update their local estimates $\hat{Q}^i$ of local q-values. Note that every $\hat{Q}^i$ is a table with $|S_i||A_i|$ components, every component corresponding to a local state-action pair $(s^i, a^i) \in S_i \times A_i$. We index all local state-action pairs by $c \in \{1, \dots, C\}$ with $C = |S_i||A_i|$, so that for the cth state action pair (s^i, a^i) we can write $\hat{Q}^i(s^i, a^i) = \hat{Q}^i_c$ the associated value in the q-table, and $\hat{Q}^i_{k,c}$ the value of that pair in table $\hat{Q}^i$ at the kth iteration of the algorithm. Then, $\hat{Q}^i(s^i, \cdot) \in \mathbb{R}^{|A_i|}$ is the vector of all possible state-action values where the local state is s^i. For every iterate we define the constraint set as the hyperrectangle $H^i = [-D, D]^C$, for a given scalar $D > 0$.

Let us now rewrite iterates (2.6) and establish their weak convergence result. For any $i \in \{1, \dots, M\}$, we first define $\bar{Q}^i_c$ the iterates in the time of each local component:

$$\bar{Q}^i_{n+1,c} = \bar{Q}^i_{n,c} + \alpha^i_n \left[r + \beta[\phi(\bar{Q}^i_{n,c})(s'^i)]^T \bar{Q}^i_k(s'^i, \cdot) - \bar{Q}^i_{n,c} \right] \tag{3.9}$$

where $\bar{Q}^i_{n,c}$ is the value of the q-table at the time of the nth update to component c, r is the reward observes at that time and s'^i the local state visited next. Note that the updates are asynchronous because every state-action pair is only updated when it is visited, *i.e.* not at every iteration of the algorithm. We can consider the "real time" as the discrete time of the algorithm: the time $\tau^i_{n,c}$ between two updates at the same component c is then the number of iterations between two visits at the corresponding local state-action pair. The iterates in real time are then:

$$\hat{Q}^i_{k+1,c} = \hat{Q}^i_{k,c} + \alpha^i_{k,c} \left[r_k + \beta[\phi(\hat{Q}^i_k)(s^i_{k+1})]^T \hat{Q}^i_k(s^i_{k+1}, \cdot) - \hat{Q}^i_{k,c} \right] I_{k,c} \tag{3.10}$$

the $(r_i, s^i_k)_k$ are collected along the agent's trajectory at every real timestep. Recall that $I_{k,c} = I_{k,(s^i,a^i)_c}$indicates that the cth local state-action pair s^i, a^i is visited at real timestep k $\alpha^i_{k,c}$ therefore takes the value $\alpha^i_{\#c}$, where $\#c$ is the number of visits to component c.

Theorem 3.3. *Consider the multi-scale Q-learning iterates (3.9). If* **A** 3.5, 3.8, *and* 3.9 *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 3.2 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** 3.7.

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 (2.4) for any update to a component c we write:

$$\pi_k = (\pi^1_k, \dots, \pi^M_k) \quad \pi^j_k = \phi(\hat{Q}^j_k) \quad \hat{Q}_k = (\hat{Q}^1_k, \dots, \hat{Q}^M_k) \quad \hat{Q}^i_k = \{\hat{Q}^i_{k,c}\}_{c=1,\dots,C} \tag{3.11}$$

We can rewrite the iterates (3.10) in real time in the following way:

$$
\begin{aligned}
\hat{Q}^i_{k+1,c} &= \hat{Q}^i_{k,c} + \alpha^i_{k,c} I_{k,c} \left[\bar{F}^i_c(\hat{Q}_k) + \delta U^i_{k,c} + \xi^i_{k,c} \right] \\
\delta U^i_{k,c} &:= Y^i_{k,c} - \mathbb{E}^Y_{k,c}[Y^i_{k,c}] \\
\xi^i_{k,c} &:= \mathbb{E}^Y_{k,c}[Y^i_{k,c}] - \bar{F}^i_c(\hat{Q}^i_k) \\
Y^i_{k,c} &:= r_k + \beta [\phi(\hat{Q}^i_k)(\cdot \mid s^i_{k+1})]^T \hat{Q}^i_k(s^i_{k+1}, \cdot) - \hat{Q}^i_{k,c} \\
&= F^i_c(\hat{Q}_k, \xi^i_{k,c}) + \delta U^i_{k,c}
\end{aligned}
\tag{3.12}
$$

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 (3.12) 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 (3.9) can be written as the multi-scale stochastic approximation iterates of Theorem 3.2.

If the noise sequences $(\xi^i_{k,c})$ satisfy **A** 3.2-3.4, then according to Theorem 3.2 the iterates follow the $M \times C$ mean ODEs:

$$
\frac{d}{dt} q^i_t(s^i_c, a^i_c) = \frac{1}{u^i_c} \bar{F}^i(q^{<i}_t, q^i_t, Z^{\geq i+1}(q^{<i+1}_t))
\tag{3.13}
$$

where $\pi^{-i} = (\phi(q^1_t), \ldots, \phi(q^{i-1}_t), \phi(q^{i+1}_t)), \ldots, \phi(q^M_t))$, the $\{q^j_t\}_{j<i}$ are constant, $\{q^j_t\}_{j>i} = Z^{>i+1}(q^{<i+1}_t)$. Then, assumption **A** 3.9 guarantees that (3.13) admits an asymptotically stable point, and we conclude on the convergence of the iterates towards a smooth equilibrium as defined in Definition 2.3.

We now need to prove that the noise sequences $\xi^i_{k,c}$ with values in the space Ξ defined in (3.12) are Markovian state-dependent noise sequences, satisfying **A** 3.2 and **A** 3.4. Let us

derive an expression for $\mathbb{E}_k^Y[Y_{k,c}^i]$. First, $\hat{Q}_{k,c}^i$ is a function of $\hat{Q}_{0,c}^i$ and the previous $Y_j^i, \tau_j^i, j < q$, so we only need to focus on s_{k+1}^i and r_k. The next state s_{k+1}^i is sampled from the local transition kernel after the agent has visited component (s^i, a^i), so we have exactly:

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

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

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

It remains to handle s_k^{-i}. It is easy to see that the state process $\{s_k^{-i}\}_k$ 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}:

$$d_{k+1} = d_k \cdot P \Pi_{j=1}^M \phi(\hat{Q}_k^j)$$

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

$$P(d_{k+1} \in \cdot | \mathcal{F}_{k,c}^{i,Y}, d_k) = P(d_{k+1} \in \cdot | \hat{Q}_k, d_k) \tag{3.14}$$

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

$$\xi_{k,c}^i = \sum_s [d_k(s|s_i) - d^{\pi_k}(s|s_i)]\left[\phi(Q_k^{-i})(\cdot, s_k^{-i})\right]^T r(s_c^i, s_k^{-i}, a_c^i, \cdot)\Big] \tag{3.15}$$

where $\pi_k = d^{\phi(Q_k)}$ is still the stationary distribution over global states under policy $\phi(Q_k)$ as defined in (2.1). Since the reward is bounded in $[-R, R]$, the $\{\xi_{k,c}^i\}_k$ take values in the compact $[-R|S||A|, R|S||A|]$. The $\{\xi_{k,c}^i\}_k$ 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 $P(\xi_{k+1,c} \in \cdot | \mathcal{F}_{k,c}^{i,Y}, d_k) = P^\xi(\xi_{k,c}, \cdot|\hat{Q}_k)$. Therefore, **A** 3.2 is satisfied.

It remains to show that the noise $\{\xi_i\}_k$ satisfies **A** 3.4: 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** 3.4, we need to prove for any compact set $A \in \Xi$,

$$\lim_{n,m} \frac{1}{m} \sum_{l=n}^{n+m-1} \mathbb{E}_n^Y\left[\xi_{l,c}(\hat{Q}) \mathbb{I}_{\{\xi_{l,c} \in A\}}\right] = 0 \tag{3.16}$$

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

$$\tilde{d}_n = d_n, \quad \tilde{d}_{k+1} = \tilde{d}_k \cdot P\Pi_{j=1}^{M} \phi(Q^j) = \tilde{d}_k \cdot P^Q \tag{3.17}$$

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 $s \in S$,

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

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 (3.17) and $d^{\phi(Q)}(s|s_i)$ the stationary distribution under policy $\phi(Q)$.

Then for all n, m, and any $\hat{Q}$ putting (3.15) into (3.16) allows us to rewrite the latter as:

$$\begin{aligned}
&\frac{1}{m}\sum_{l=n}^{n+m-1} \mathbb{E}_n^Y \left[\xi_{l,c}(\hat{Q}) \mathbb{I}_{\{\xi_n \in A\}}\right] = \frac{1}{m}\sum_{l=n}^{n+m-1} \left[\left(\tilde{D}_l(\hat{Q}) - D(\hat{Q})\right) R_{\pi_l}\right] \mathbb{I}_{\{\xi_l \in A\}} \\
&= \frac{1}{m}\sum_{l=n}^{n+m-1} \left[\left((\tilde{D}_n(\hat{Q})\Pi_{j=n}^{l-1} P^{\hat{Q}}) - D(\hat{Q})\right) R_{\pi_l}\right] \mathbb{I}_{\{\xi_l \in A\}} \\
&= \frac{1}{m}\sum_{l=n}^{n+m-1} \left[\left(d_n \left(P^{\hat{Q}}\right)^{l-n} - D(\hat{Q})\right) R_{\pi_l}\right] \mathbb{I}_{\{\xi_l \in A\}} \\
&\leq \frac{R}{m}\sum_{l'=0}^{m-1} \left\| \left(d_n \left(P^{\hat{Q}}\right)^{l'} - D(\hat{Q})\right) \right\|_1 \mathbb{I}_{\{\xi_{l'+n} \in A\}}
\end{aligned} \tag{3.18}$$

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 [23], 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 (3.18) we have that:

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

Among the assumptions under which the convergence of the multi-scale Q-learning iterates is guaranteed, **A** 3.5 and 3.8 set us constraints on the learning rate sequences, and **A** 3.9 posits the existence of solutions for the mean ODEs approximating the q-iterates. In the next section, we will zoom in on **A** 3.9 to understand the set of transition independent Dec-POMDPs that can satisfy it.

4. Multi-scale Q-learning iterates for a special dec-POMDP structure. In this section we will introduce and discuss structures of Dec-POMDPs that can satisfy **A** 3.9. First, following Theorem 3.3, under what circumstances can we attribute M different learning rate sequences to M different agents so that **A** 3.9 is verified ? 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 Subsection 4.1, we will start by making explicit what is meant by ordering agents according to their dynamics through **A** 4.1. 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, Subsection 4.2 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.

4.1. 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 (3.11), 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 (2.4). 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 (2.6) will have a single fixed point. We will now formalize this condition.

A 4.1. *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 4.1. *Let us consider M agents locally updating their q-values estimates according to (3.9) with initial values $\hat{Q}_0^i \in [-D, D]$ for $D > 0$ such that $D > \frac{R}{\beta}$. Suppose that **A** 4.1 is satisfied with the ordering of agents $\{1, \dots, M\}$, and the learning rates $\{\alpha_i\}_{1 \dots M}$ follow **A** 3.5*

and **A** 3.8*, 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.*

Proof. We recall the mean ODE followed by each agent i as introduced in (3.13):

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

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** 4.1 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** 3.9 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** 3.5 and 3.8 are satisfied. The weak convergence of the iterates $\hat{Q}_k$ towards a smoothed equilibrium then follows from Theorem 3.3. ∎

This learning rates attribution however forces us to have as many learning rates as we have agents. We notice that Subsection 4.1 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.

4.2. 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 [19] 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** 3.5, 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 4.1 is preserved if every agent i is assigned the learning rate sequence $\alpha_k^{rk(i)}$. Let us start by rewriting **A** 4.1 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:

A 4.2. *For every agent i in $\mathcal{G}$ and with the same notations as **A** 4.1, 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 3.3.

- (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 D 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 5.1.

As an example, we will consider a certain type of graph structure for which a very simple procedure can return a ranking satisfying (A) and (B). To further expose the problem, we highlight 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 $\mathcal{G}\}$. We will refer to the learning rate assignation procedure for this subset of graph as `TreeLRs`: it attributes to every node the size of the longest path from a node without any incoming edge. Because the graph is finite and acyclic, there is a set of nodes $\mathcal{V}_0 \subset \mathcal{V}$ of out-degree 0, corresponding to a set of agents influencing no other agent, and a set of nodes $\mathcal{V}_R \subset \mathcal{V}$ of indegree 0, corresponding to a set of agents influenced by no other agent. Now for any pair of nodes i, j let us define $\mathcal{W}(i, j)$ the set of paths from node i to node j. Any path p in $\mathcal{W}(i, j)$ has a length $|p|$, defined

as the number of edges in the path. The graph is finite, so $\mathcal{W}(i,j)$ is either empty or there is $d(P(i,j)) = \max_{p \in \mathcal{W}(i,j)} |p| \geq 0$ the maximum length of any path from i to j. For a given node i, `TreeLRs` writes $\mathcal{P}(\mathcal{V}_R, i) = \{\mathcal{P}(j,i) \mid j \in \mathcal{V}_R\}$ the set of paths from any node in $\mathcal{V}_R$ to i. $\mathcal{P}(\mathcal{V}_R, i)$ must be non-empty, and thus we define $d^i = d(\mathcal{P}(\mathcal{V}_R, i))$ as the level of i. If $i \in \mathcal{V}_R$ then $d^i = 0$. The attribution of learning rates under `TreeLRs` is then as follows.

First, `TreeLRs` assigns to every agent $i \in \{1, \dots, M\}$ in the environment the level of its corresponding node d^i. This returns $\bar{M} < M$ different levels. Let us have $\bar{M}$ learning rate sequences α_k^j for $j \in \{1, \dots, \bar{M}\}$ satisfying **A** 3.5 and 3.8, such that for every pair $j > l$, $\frac{\alpha_k^l}{\alpha_k^j} \to \infty$ when $k \to \infty$. Then, `TreeLRs` chooses $rk(i) = d^i$, and attribute to each agent i the corresponding learning rate sequence $\alpha_k^{rk(i)}$. We now show that under this structure and learning rates attribution, convergence to a global equilibrium is preserved when every agent only receives a local reward gathered from its neighbors.

Theorem 4.2. *Let us consider M agents locally updating their q-values estimates according to iterates*

$$\hat{Q}^i_{k+1,c} = \hat{Q}^i_{k,c} + \alpha^i_k(s^i_k, a^i_k)\left[\bar{r}^i_k + \beta[\phi(\hat{Q}^i_k)(s^i_{k+1})]^T \hat{Q}^i_k(s^i_{k+1}, \cdot) - \hat{Q}^i_{k,c}\right] \tag{4.1}$$

*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** 4.2. Let the learning rates $\{\alpha_k\}$ be attributed by a ranking that satisifes (A) and (B). Then the conclusion of Theorem 4.1 stands.*

The iterates (4.1) will be labeled `NetworkMQL`. The proof is detailed in Appendix E. We start by showing any ranking satisfying (A) and (B) preserves the convergence of Theorem 3.3 and then that `TreeLRs` returns a learning rate distributions that belongs to that set.

5. Application to wind farm control. We evaluate the performance of our multi-scale approach on a Dec-POMDP experiment by considering a problem from the industry: wind farm control.

Wind turbines are often grouped together on the same field in what are known as wind farms. Yet an operating wind turbine causes local wind perturbations - called wake effects - that can reduce the production of its neighbors. The angle between any turbine's rotor and the direction of the wind, called yaw, can be increased to diminish the impact of the perturbations on its neighbors.

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

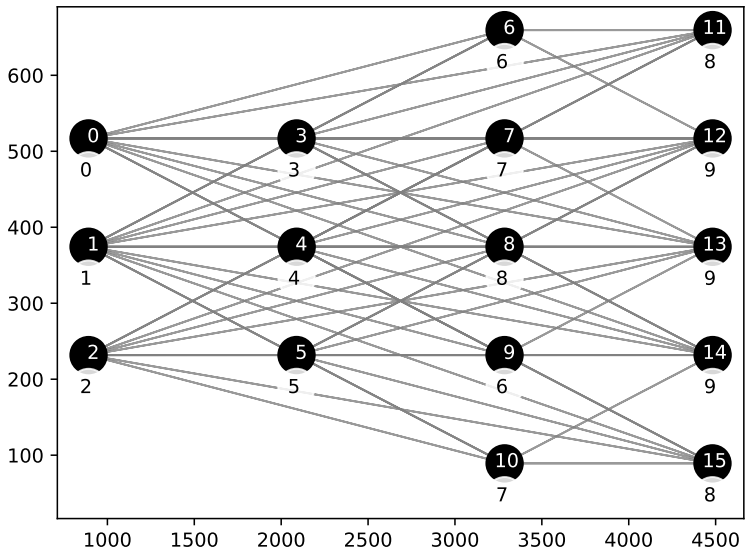


Figure 5.1: 16 interacting wind turbines on a graph. The levels used in the MQL algorithm are written in white, and the corresponding levels used in the NetworkMQL algorithm obtaines with Algorithm D.1 are written in black. The coordinates represent the location of each turbine in the farm.

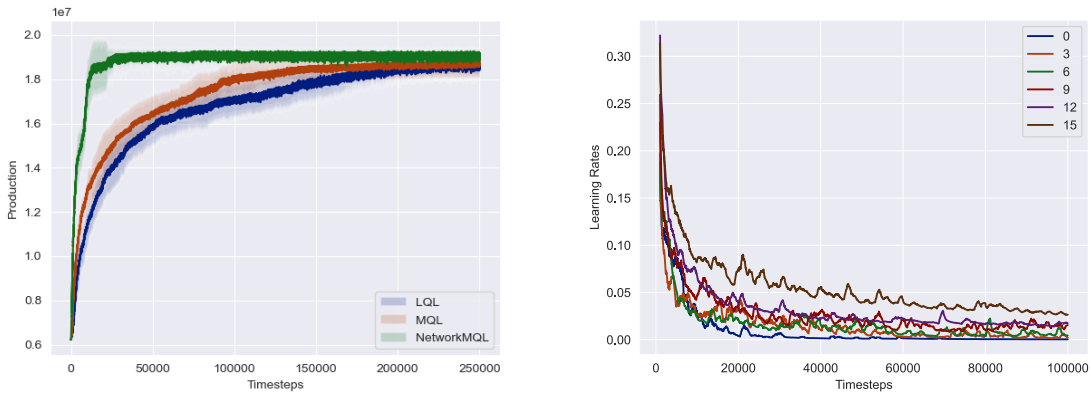
wind conditions, the problem is not transition-independent: any action taken by an agent can change the wind conditions at other agent's locations. This can be fixed by using a direct observation of W as wind statistics in the local state.

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 $0 < l_{\bar{M}} < \cdots < l_1 < 1$ 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



(a) Evolution of power output

(b) Evolution of learning rates

Figure 5.2: MQL: Multi-scale. NetworkMQL: Multi-scale with Reward Decomposition LQL: Local Q-learning. The total power output of the simulated 16 turbines wind farm averaged on 1000 time-steps is reported on Figure 5.2a. The evolution of learning rates under MQL based on scales attributed in Figure 5.1 is reported on Figure 5.2b for the first $100k$ time-steps.

approximate learning rates that satisfy **A** 3.8 by adding a multiscale term dependent on the time between visits, so that the final learning rate sequences are:

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

where $T_{k,c}^i$ is as before the real time of the kth update to component c. We use the same gain $g = 2$ for all algorithms. An example of the evolution of these learning rates for Algortihm *MQL* can be found on Figure 5.2b. We run both Algorithm *MQL* (3.9) and Algorithm *NetworkMQL* (4.1) 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 5.2a. For *MQL*, we simply assign a different rank to every agent following a topological sort and use the M multiscale learning rate sequences $\alpha_{k,c}^{\prime l_i}$. 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 D to assign $\bar{M} \leq M$ ranks to all agents in the DAG. We obtain $\bar{M} = 9$ different ranks shown in Figure 5.1 and use the last 9 learning rate sequences in $\{l_i\}_{i\in 1\dots M}$.

6. 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. To provide a first step towards bridging

this gap, we have here focused on transition-independent Dec-PODMP, and shown that in these problems the partial observability of the global state can be modeled as a Markovian perturbation in a stochastic approximation iterates. We have shown 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 have then applied these results to wind farm control, a real optimization problem from the industry.

Further work can 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 ([34, 31]), and our acyclic dependence analysis could provide a basis to find a theoretical explanation of these results.