# Reinforcement Learning in Decentralized Stochastic Control Systems with Partial History Sharing

Jalal Arabneydi[1] and Aditya Mahajan[2]

*Abstract*— **In this paper, we are interested in systems with multiple agents that wish to collaborate in order to accomplish a common task while a) agents have different information (decentralized information) and b) agents do not know the model of the system completely i.e., they may know the model partially or may not know it at all. The agents must learn the optimal strategies by interacting with their environment i.e., by decentralized Reinforcement Learning (RL). The presence of multiple agents with different information makes decentralized reinforcement learning conceptually more difficult than centralized reinforcement learning. In this paper, we develop a decentralized reinforcement learning algorithm that learns $\epsilon$-team-optimal solution for partial history sharing information structure, which encompasses a large class of decentralized control systems including delayed sharing, control sharing, mean field sharing, etc. Our approach consists of two main steps. In the first step, we convert the decentralized control system to an equivalent centralized POMDP (Partially Observable Markov Decision Process) using an existing approach called common information approach. However, the resultant POMDP requires the complete knowledge of system model. To circumvent this requirement, in the second step, we introduce a new concept called "Incrementally Expanding Representation" using which we construct a finite-state RL algorithm whose approximation error converges to zero exponentially fast. We illustrate the proposed approach and verify it numerically by obtaining a decentralized Q-learning algorithm for two-user Multi Access Broadcast Channel (MABC) which is a benchmark example for decentralized control systems.**

## 1. INTRODUCTION

### A. Motivation

Decentralized decision making is relevant in a wide range of applications ranging from networked control systems, robotics, transportation networks, communication networks, sensor networks, and economics. There is a rich history of research on optimal stochastic control of decentralized system. We refer the reader to [1] for a detailed review.

Most of the literature assumes that the system model is completely known to all decision makers; however, in practice, such knowledge may only be available partially or may not be available. Hence, it is crucial for decision makers to be able to learn the optimal solutions. In the literature, learning in centralized stochastic control is well studied and there exist many approaches such as model-predictive control, adaptive control, and reinforcement learning. This is in contrast to the learning in decentralized stochastic control; it is not immediately clear on how centralized learning approaches would work for decentralized systems. In this paper, we propose a novel Reinforcement Learning (RL) algorithm for a class of decentralized stochastic control systems that guarantees team-optimal solution.

Existing approaches for multi-agent learning may be categorized as follows: exact methods and heuristics. The exact methods rely on the assumption that the information structure is such that all agents can consistently update the Q-function. These include approaches that rely on social convention and rules to restrict the decisions made by the agents [10]; approaches that use communication to convey the decisions to all agents [11]; and approaches that assume that the Q-function decomposes into a sum of terms, each of which is independently updated by an agent [12]. Heuristic approaches include joint action learners heuristic [13], where each agent learns the empirical model of the system in order to estimate the control action of other agents; frequency maximum Q-value heuristic [14], where agents keep track of the frequency with which each action leads to a "good" outcome; heuristic Q-learning [15], which assigns a rate of punishment for each agent; and distributed Q-learning [16], which uses predator-prey models to assign heuristic sub-goals to individual agents. To the best of our knowledge, there is no RL approach that guarantees team-optimal solution. In this paper, we present such an approach.

We describe the system model and problem formulation in Sections 1.3 and 1.4, respectively. We state the main challenges in Section 1.5 and our main contributions in Section 1.6. In Section 2, we present a brief preliminary on Partial History Sharing (PHS) information structure. We describe our approach in two basic steps in Section 3. In Section 4, we mention a few points on the implementation. Based on the proposed approach, we develop a RL algorithm for a benchmark example with numerical results in Section 5.

### B. Notation

We use upper-case letters to denote random variables (e.g. $X$) and lower-case letters to denote their realizations (e.g. $x$). We use the short-hand notation $X_{a:b}$ for the vector $(X_a, X_{a+1}, \dots, X_b)$ and bold letters to denote vectors e.g. $\mathbf{Y} = (Y^1, \dots, Y^n)$. $\mathbb{P}(\cdot)$ is the probability of an event, $\mathbb{E}[\cdot]$ is the expectation of a random variable, and $|\cdot|$ is the absolute value of a real number. $\mathbb{N}$ refers to the set of natural numbers and $\mathbb{Z}^+ = \mathbb{N} \cup \{0\}$.

### C. System Model

Let $X_t \in \mathcal{X}$ denote the state of a dynamical system controlled by $n$ agents. At time $t$, agent $i$ observes $Y_t^i \in \mathcal{Y}^i$ and

[1]Jalal Arabneydi is with the Department of Electrical Engineering, McGill University, 3480 University St., Montreal, QC, Canada jalal.arabneydi@mail.mcgill.ca

[2]Aditya Mahajan is with the Department of Electrical Engineering, McGill University, 3480 University St., Montreal, QC, Canada aditya.mahajan@mcgill.ca

chooses $U_t^i \in \mathcal{U}^i$. For ease of notation, we denote the joint actions and the joint observations by $\mathbf{U}_t = (U_t^1, \dots, U_t^n) \in \mathcal{U}$ and $\mathbf{Y}_t = (Y_t^1, \dots, Y_t^n) \in \mathcal{Y}$, respectively. The dynamics of the system are given by

$$X_{t+1} = f(X_t, \mathbf{U}_t, W_t^s), \tag{1}$$

and the observations are given by

$$\mathbf{Y}_t = h(X_t, \mathbf{U}_{t-1}, W_t^o), \tag{2}$$

where $\{W_t^s\}_{t=1}^\infty$ is an i.i.d. process with probability distribution function $P_{W^s}$, $\{W_t^o\}_{t=1}^\infty$ is an i.i.d. process with probability distribution function $P_{W^o}$, and $X_1$ is the initial state with probability distribution function $P_X$. The primitive random variables $\{X_1, \{W_t^s\}_{t=1}^\infty, \{W_t^o\}_{t=1}^\infty\}$ are mutually independent and defined on a common probability space.

For ease of exposition, we assume all system variables are finite valued. Let $I_t^i \subseteq \{\mathbf{Y}_{1:t}, \mathbf{U}_{1:t-1}\}$ be information available at agent $i$ at time $t$. The collection $(\{I_t^i\}_{t=1}^\infty, i = 1, \dots, n)$ is called the *information structure*. In this paper, we restrict attention to an information structure called *partial history sharing* [2], which will be defined later.

At time $t$, agent $i$ chooses action $U_t^i$ according to *control law* $g_t^i$ as follows

$$U_t^i = g_t^i(I_t^i). \tag{3}$$

We denote $\boldsymbol{g}^i = (g_1^i, g_2^i, \dots)$ as *strategy* of agent $i$ and $\boldsymbol{g} = (\boldsymbol{g}^1, \dots, \boldsymbol{g}^n)$ as joint strategy of all the agents. The performance of strategy $\mathbf{g}$ is measured by the following infinite-horizon discounted cost

$$J(\boldsymbol{g}) = \mathbb{E}^{\boldsymbol{g}}\left[\sum_{t=1}^{\infty} \beta^{t-1} \ell(X_t, \mathbf{U}_t)\right], \tag{4}$$

where $\beta \in (0,1)$ is the discount factor, $\ell$ is the per-step cost function, and the expectation is with respect to a joint probability distribution on $(X_{1:\infty}, \mathbf{U}_{1:\infty})$ induced by the joint probability distribution on the primitive random variables and the choice of strategy $\boldsymbol{g}$.

A strategy $\boldsymbol{g}^*$ is optimal if for any other strategy $\boldsymbol{g}$, $J(\boldsymbol{g}^*) \leq J(\boldsymbol{g})$. For $\epsilon > 0$, strategy $\boldsymbol{g}^*$ is $\epsilon$-optimal, if for any other strategy $\boldsymbol{g}$, $J(\boldsymbol{g}^*) \leq J(\boldsymbol{g}) + \epsilon$.

### D. Problem Formulation

We will consider three different setups that differ in the assumptions about the knowledge of the model. For all the setups, we will assume that the action and the observation spaces as well as the information structure, the discount factor $\beta$, and an upper-bound on the per-step cost are common knowledge between all agents. The setups differ in the assumptions about state space $\mathcal{X}$, system dynamics and observations $(f, h)$, probability distributions $(P_X, P_{W^s}, P_{W^o})$, and cost structure $\ell$. These include two setups, 1) complete-knowledge of the model, and 2) incomplete-knowledge of the model which includes two sub-cases: 2a) partial-knowledge of the model and 2b) no-knowledge of the model.

In general, the complete-knowledge of the model is required to find an optimal strategy $\boldsymbol{g}^*$. However, in practice, there are many applications where such information is not completely available or is not available at all. In such applications, the agents must learn the optimal strategy by interacting with their environment. This is known as reinforcement learning (RL). If the agents have partial knowledge of the model, the setup is called *model-based* RL. If the agents have no knowledge of the model, setup is called *model-free* RL.

Define $L := \max_{x,\mathbf{u}} |\ell(x, \mathbf{u})|$. We are interested in the following problem.

**Problem 1** *Given the information structure, action spaces $\{\mathcal{U}^i\}_{i=1}^n$, observation spaces $\{\mathcal{Y}^i\}_{i=1}^n$, discount factor $\beta$, the upper-bound $L$ on per-step cost, and any $\epsilon > 0$, develop a (model-based or model-free) reinforcement learning algorithm using which the agents learn an $\epsilon$-optimal strategy $\boldsymbol{g}^*$.*

### E. Main Difficulties

Given the complete knowledge of system model, finding team-optimal solution in decentralized control systems is conceptually challenging due to the decentralized nature of information available to the agents. The agents need to cooperate with each other to fulfill a common objective while they have different perspectives about themselves, other agents, and the environment. This discrepancy in perspectives makes establishing cooperation among agents difficult; we refer reader to [21] for details. Thus, finding team-optimal solution is even more challenging when agents have only partial knowledge or no knowledge of system model. Hence, it is difficult to *consistently* learn strategies in such settings.

### F. Contributions

Below, we mention our main contributions in this paper.

1) We propose a novel approach to perform reinforcement learning in a large class of decentralized stochastic control systems with partial history sharing (PHS) information structure that guarantees $\epsilon$-team-optimal solution. In particular, our approach combines the *common information approach* of [2] with any RL algorithm of Partially Observable Markov Decision Processes (POMDP). The approach works in two steps. In the first step, the common information approach is used to convert the decentralized control problem to an equivalent centralized POMDP and in the second step, a RL algorithm is used to provide a learning scheme to identify an $\epsilon$-optimal strategy in the resultant POMDP. Note that any RL algorithm of POMDPs may be used in the second step; however, we develop a new methodology for the second step as explained below.

2) We propose a novel methodology to perform reinforcement learning in *centralized* POMDPs as an intermediate step of the two-step approach described above. (This methodology by itself may be of interest due to the fact that developing RL algorithm in POMDPs is difficult). The methodology consists of three parts: 1) converting the POMDP to a countable-state MDP $\Delta$ by defining a new concept that we call Incrementally Expanding Representation (IER), 2) approximating $\Delta$ with a sequence of finite-state MDPs $\{\Delta_N\}_{N=1}^\infty$, and 3) using a RL algorithm to learn an optimal strategy of MDP $\Delta_N$. We show that the performance of the RL strategy converges to the optimal performance exponentially as $N \to \infty$. We use this methodology in the second step of the two-step approach.

3) Using the proposed two-step approach, we develop a RL algorithm for two-user Multi Access Broadcast Channel (MABC) which is used as a benchmark for decentralized control systems. Numerical simulations validate that the RL algorithm converges to an optimal strategy.

## 2. Preliminaries on Partial History Sharing

Herein, we present a simplified version of partial history sharing information structure, originally presented in [2].

**Definition 1 ([2], Partial History Sharing (PHS))** *Consider a decentralized control system with $n$ agents. Let $I_t^i$ denote the information available to agent $i$ at time $t$. Assume $I_t^i \subseteq I_{t+1}^i$. Then, split the information at each agent into two parts: common information $C_t = \bigcap_{i=1}^N I_t^i$ i.e. the information shared between all agents and local information $M_t^i = I_t^i \backslash C_t$ that is the local information of agent $i$. Define $Z_t := C_{t+1} \backslash C_t$ as common observation, then $C_{t+1} = Z_{1:t}$. An information structure is called partial history sharing when the following conditions are satisfied:*

*a) The update of local information*

$$M_{t+1}^i \subseteq \{M_t^i, U_t^i, Y_{t+1}^i\} \backslash Z_t, \quad i \in \{1, \dots, n\}.$$

*b) For every agent $i$, the size of the local information $M_t^i$ and the size of the common observation $Z_t$ are uniformly bounded in time $t$.*

These conditions are fairly mild and are satisfied by a large class of models. Examples include delayed sharing [17], periodic sharing [18], mean-field sharing [19], etc. Even for models that do not satisfy the above conditions directly, it is often possible to identify sufficient statistics that satisfy the above conditions, e.g., control sharing [20].

**Remark 1** *Note that the conditions (a) and (b) are valid even if there is no common information between agents i.e., $C_t = \emptyset$. Hence, the decentralized control systems with pure decentralized information (i.e. no information commonly shared) falls into PHS information structure.*

## 3. Approach

In this part, we derive a RL algorithm for systems with PHS information structure. Our approach consists of two steps. In the first step, we consider the setup of the complete-knowledge of the model and use the *common information approach* of [2] to convert the decentralized control problem to an equivalent centralized POMDP. In the second step, we consider the setup of incomplete-knowledge of the model and develop a finite-state RL algorithm based on the POMDP obtained in the first step.

### A. Step 1: An Equivalent Centralized POMDP

In this section, we present common information approach of [2] and its main results for the setup of complete-knowledge of the model described in Section 1.3.

Let $\mathcal{M}^i$ and $\mathcal{Z}$ denote the spaces of realizations of local information of agent $i$ and common observation, respectively. Consider a virtual *coordinator* that observes the common information $C_t$ shared between all agents and chooses $(\Gamma_t^1, \dots, \Gamma_t^n)$, where $\Gamma_t^i : \mathcal{M}^i \mapsto \mathcal{U}^i$ is the mapping from the local information of agent $i$ to action of agent $i$ at time $t$, according to

$$\Gamma_t^i = \psi_t^i(C_t), \quad i \in \{1, \dots, n\}. \tag{5}$$

We call $\boldsymbol{\psi}_t := \{\psi_t^1, \dots, \psi_t^n\}$ the *coordination law* and $\boldsymbol{\Gamma}_t = (\Gamma_t^1, \dots, \Gamma_t^n)$ the *prescription*. The agents use this prescription to choose their actions as follows:

$$U_t^i = \Gamma_t^i(M_t^i), \quad i \in \{1, \dots, n\}. \tag{6}$$

We denote the space of mappings $\Gamma_t^i$ by $\mathcal{G}^i$ and the space of prescriptions $\boldsymbol{\Gamma}_t$ by $\mathcal{G} = \prod_{i=1}^n \mathcal{G}^i$. In the sequel, for ease of notation, we will use the following compact form for the coordinator's law,

$$\boldsymbol{\Gamma}_t = \boldsymbol{\psi}_t(C_t). \tag{7}$$

We call $\boldsymbol{\psi} = \{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \dots\}$ as the *coordination strategy*. In the *coordinated system*, dynamics and cost function are as same as those in the original problem in Section 1.3. In particular, the infintie-horizon discounted cost in the coordinated system is as follows:

$$J(\boldsymbol{\psi}) = \mathbb{E}^{\boldsymbol{\psi}}\left[\sum_{t=1}^{\infty} \beta^{t-1} \ell(\mathbf{X}_t, \boldsymbol{\Gamma}_t^1(M_t^1), \dots, \Gamma_t^n(M_t^n))\right]. \tag{8}$$

**Lemma 1 ([2], Proposition 3)** *The original system described in Section 1.3 with PHS information structure is equivalent to the coordinated system.*

We denote $\boldsymbol{M}_t = (M_t^1, \dots, M_t^n)$ as the joint local information. According to [2], $\Pi_t = \mathbb{P}(X_t, \mathbf{M}_t | Z_{1:t-1}, \boldsymbol{\Gamma}_{1:t-1})$ is an information state for the coordinated system. It is shown in [2] that:

1) There exists a function $\phi$ such that
$$\Pi_{t+1} = \phi(\Pi_t, \boldsymbol{\Gamma}_t, Z_t). \tag{9}$$
2) The observation $Z_t$ only depends on $(\Pi_t, \Gamma_t)$ i.e.
$$\mathbb{P}(Z_t | \Pi_{1:t}, \boldsymbol{\Gamma}_{1:t}) = \mathbb{P}(Z_t | \Pi_t, \boldsymbol{\Gamma}_t). \tag{10}$$
3) There exists a function $\hat{\ell}$ such that
$$\hat{\ell}(\pi_t, \boldsymbol{\gamma}_t) = \mathbb{E}[\ell(X_t, \mathbf{U}_t | Z_{1:t-1} = z_{1:t-1}, \boldsymbol{\Gamma}_{1:t} = \boldsymbol{\gamma}_{1:t})]. \tag{11}$$

Assume that the initial state $\pi_1$ is fixed. Let $\mathcal{R}$ denote the reachable set of above centralized POMDP that contains all the realizations of $\pi_t$ generated by $\pi_{t+1} = \phi(\pi_t, \boldsymbol{\gamma}, z), \forall \boldsymbol{\gamma} \in \mathcal{G}, \forall z \in \mathcal{Z}, \forall t \in \mathbb{N}$, with initial information state $\pi_1$. Note that since all the variables are finite valued, then $\mathcal{G}$ (set of all prescriptions $\boldsymbol{\gamma}$) and $\mathcal{Z}$ (set of all observations of the coordinator) are finite sets. Hence, $\mathcal{R}$ is at most a countable set.

**Theorem 1 ([2], Theorem 5)** *Let $\boldsymbol{\psi}^*(\pi)$ be any argmin of the right-hand side of following dynamic program. For $\pi \in \mathcal{R}$,*

$$V(\pi) = \min_{\boldsymbol{\gamma}} (\hat{\ell}(\pi, \boldsymbol{\gamma}) + \beta \mathbb{E}[V(\phi(\pi, \boldsymbol{\gamma}, Z_t)) | \Pi_t = \pi, \boldsymbol{\Gamma}_t = \boldsymbol{\gamma}]),$$

*where $\boldsymbol{\gamma} = (\gamma^1, \dots, \gamma^n)$ and the minimization is over all functions $\gamma^i \in \mathcal{G}^i, i \in \{1, \dots, n\}$. Then, the joint stationary strategy $\boldsymbol{g}^* = (\boldsymbol{g}^{1,*}, \dots, \boldsymbol{g}^{n,*})$ is optimal such that*

$$g^{i,*}(\pi, m^i) := \boldsymbol{\psi}^{i,*}(\pi)(m^i), \quad \pi \in \mathcal{R}, m^i \in \mathcal{M}^i, \forall i.$$

In the next step, we develop a finite-state RL algorithm based on the obtained POMDP for the setup of incomplete-knowledge of the model.

### *B. Step 2: Finite-State RL Algorithm For POMDP*

In the previous step, we identified a centralized POMDP that is equivalent to the decentralized control system with PHS information structure. However, the obtained POMDP requires the complete knowledge of the model. To circumvent this requirement, we introduce a new concept that we call *Incrementally Expanding Representation* (IER). The main feature of IER is to remove the dependency of the POMDP from the complete knowledge of the model. Based on a proper IER, in this step, we develop a finite-state RL algorithm. This step consists of three parts. In part (1), we convert the POMDP to a countable-state MDP $\Delta$ without loss of optimality. In part (2), we construct a sequence of finite-state MDPs $\{\Delta_N\}_{N=1}^{\infty}$ of MDP $\Delta$. In part (3), we use a generic RL algorithm to learn an optimal strategy of $\Delta_N$.

**Definition 2 (Incrementally Expanding Representation)** *Let $\{\mathcal{S}_k\}_{k=1}^{\infty}$ be a sequence of finite sets such that $\mathcal{S}_1 \subsetneq \mathcal{S}_2 \subsetneq \ldots \subsetneq \mathcal{S}_k \subsetneq \ldots$, and $\mathcal{S}_1$ is a singleton, say $\mathcal{S}_1 = \{s^*\}$. Let $\mathcal{S} = \lim_{k\to\infty} \mathcal{S}_k$ be the countable union of above finite sets, $B : \mathcal{S} \to \mathcal{R}$ be a surjective function that maps $\mathcal{S}$ to the reachable set $\mathcal{R}$, and $\tilde{f} : \mathcal{S}\times\mathcal{G}\times\mathcal{Z} \to \mathcal{S}$. The tuple $\langle\{\mathcal{S}_k\}_{k=1}^{\infty}, B, \tilde{f}\rangle$ is called an* Incrementally Expanding Representation *(IER), if it satisfies the following properties:*

*(P1) Incremental Expansion: For any $\boldsymbol{\gamma} \in \mathcal{G}, z \in \mathcal{Z}$, and $s \in \mathcal{S}_k$, we have that*

$$\tilde{f}(s, \boldsymbol{\gamma}, z) \in \mathcal{S}_{k+1}. \tag{12}$$

*(P2) Consistency: For any $(\boldsymbol{\gamma}_{1:t-1}, z_{1:t-1})$, let $\pi_t$ and $s_t$ be the states obtained by recursive application of* (9) *and* (12) *starting from $\pi_1$ and "$s^*$, respectively. Then,*

$$\pi_t = B(s_t). \tag{13}$$

In general, every decentralized control system with PHS information structure has at least one IER. In the following example, we present a generic IER that is valid for every system with PHS information structure.

*Example 1:* Let $S_1 = \{\emptyset\}$, $S_2 = \{\emptyset\} \cup \{\mathcal{G} \times \mathcal{Z}\}$, and $S_{k+1} = S_k \cup \{\mathcal{G} \times \mathcal{Z}\}^k, k \in \mathbb{N}$. Let $S = \lim_{k\to\infty} S_k$ and $B : \mathcal{S} \to \mathcal{R}$ such that

$$B(\emptyset) = \pi_1, B(s_{k+1}) = \phi(\phi(\ldots., \boldsymbol{\gamma}_{k-1}, z_{k-1}), \boldsymbol{\gamma}_k, z_k) = \pi_{k+1},$$

where $s_{k+1} = ((\boldsymbol{\gamma}_1, z_1), \ldots, (\boldsymbol{\gamma}_k, z_k)) \in \mathcal{S}_{k+1}$. Define $\tilde{f}$ as follows:

$$\tilde{f}(s, \boldsymbol{\gamma}, z) = s \circ \boldsymbol{\gamma} \circ z,$$

where $\circ$ denotes concatenation. By construction, tuple $\langle\{\mathcal{S}_k\}_{k=1}^{\infty}, B, \tilde{f}\rangle$ satisfies (P1) and (P2), and hence is an IER.

*1) Countable-state MDP $\Delta$:* Let the tuple $\langle\{\mathcal{S}_k\}_{k=1}^{\infty}, B, \tilde{f}\rangle$ be an IER of the POMDP obtained in the first step. Then, define MDP $\Delta$ with countable state space $\mathcal{S}$, finite action space $\mathcal{G}$, and dynamics $\tilde{f}$ such that:

**(F1)** The initial state is singleton $s^*$. The state $S_t \in \mathcal{S}_k$, $k \le t$, evolves as follows: for $\boldsymbol{\Gamma}_t \in \mathcal{G}, Z_t \in \mathcal{Z}$,

$$S_{t+1} = \tilde{f}(S_t, \boldsymbol{\Gamma}_t, Z_t), \quad S_{t+1} \in \mathcal{S}_{k+1} \tag{14}$$

where observation $Z_t$ only depends on $(S_t, \boldsymbol{\Gamma}_t)$ (that is a consequence of (10) and consistency property in (13)). At time $t$, there is a cost depending on the current state $S_t \in \mathcal{S}$ and action $\boldsymbol{\Gamma}_t \in \mathcal{G}$ given by

$$\tilde{\ell}(S_t, \boldsymbol{\Gamma}_t) := \hat{\ell}(B(S_t), \boldsymbol{\Gamma}_t) = \hat{\ell}(\Pi_t, \boldsymbol{\Gamma}_t). \tag{15}$$

**(F2)** State space $\mathcal{S}$, action space $\mathcal{G}$, and dynamics $\tilde{f}$ do not depend on the unknowns.

The performance of a stationary strategy $\tilde{\boldsymbol{\psi}} : \mathcal{S} \mapsto \mathcal{G}$ is quantified by

$$\tilde{J}(\tilde{\boldsymbol{\psi}}) = \mathbb{E}^{\tilde{\boldsymbol{\psi}}}\left[\sum_{t=1}^{\infty} \beta^{t-1}\tilde{\ell}(S_t, \boldsymbol{\Gamma}_t)\right]. \tag{16}$$

There may exist more than one IER that satisfy above features. For instance, the IER of Example 1 always satisfies (F1) and (F2) (that is model-free). This IER can also be used in the model-based cases; however, in the model-based cases, due to having partial knowledge of the model, one may be able to find a simpler IER. See Section 5 for an example.

**Lemma 2** *Let $\tilde{\boldsymbol{\psi}}^*$ be an optimal strategy for MDP $\Delta$. Construct a strategy $\boldsymbol{\psi}^*$ for the coordinated system as follows:*

$$\tilde{\boldsymbol{\psi}}^*(s) =: \boldsymbol{\psi}^*(B(s)), \quad \forall s \in \mathcal{S}. \tag{17}$$

*Then, $\tilde{J}(\tilde{\boldsymbol{\psi}}^*) = J(\boldsymbol{\psi}^*)$ and $\boldsymbol{\psi}^*$ is an optimal strategy for the coordinated system, and therefore can be used to generate an optimal strategy for the decentralized control system.*

Proof is omitted due to lack of space.

*2) Finite-state incrementally expanding MDP $\Delta_N$ :* In this part, we construct a series of finite-state MDPs $\{\Delta_N\}_{N=1}^{\infty}$, that approximate the countable-state MDP $\Delta$ as follows. Let $\Delta_N$ be a finite-state MDP with state space $\mathcal{S}_N$ and action space $\mathcal{G}$. The transition probability of $\Delta_N$ is constructed as follows. Pick any arbitrary set $D^* \in \mathcal{S}_N$. Remap every transition in $\Delta$ that takes the state $s \in \mathcal{S}_N$ to $s' \in \mathcal{S}_{N+1}\backslash\mathcal{S}_N$ to a transition from $s \in \mathcal{S}_N$ to any (not necessarily unique) state in $D^*$. In addition, the per-step cost function of $\Delta_N$ is simply a restriction of $\tilde{\ell}$ to $\mathcal{S}_N \times \mathcal{G}$.

We assume that there exists an action or a sequence of actions that if taken, the system transmits to a known state $d^*$ in $D^*$. For example, suppose there is a reset action in the system. After executing the reset action, the state of the system is reset and transmitted to a known state $d^* \in D^*$. Let $\tau_N \in \mathbb{N}$ be the longest amount of time during which $S_t$, $t \le \tau_N$, stays in $\mathcal{S}_N$ under dynamics $\tilde{f}$, optimal strategy $\tilde{\boldsymbol{\psi}}^*$, and any arbitrary sample path of $z_{1:\tau_N-1}$, i.e.,

$$S_t = \tilde{f}(S_{t-1}, \tilde{\boldsymbol{\psi}}^*(S_{t-1}), Z_{t-1}) \in \mathcal{S}_N, \quad \forall t \le \tau_N. \tag{18}$$

Let $\tilde{\boldsymbol{\psi}}_N^*$ and $\tilde{J}_N(\tilde{\boldsymbol{\psi}}_N^*)$ be an optimal stationary strategy of $\Delta_N$ and the optimal cost (performance) of $\Delta_N$, respectively.

**Theorem 2** *The difference in performance between $\Delta$ and $\Delta_N$ is bounded as follows:*

$$|\tilde{J}(\tilde{\boldsymbol{\psi}}^*) - \tilde{J}_N(\tilde{\boldsymbol{\psi}}_N^*)| \le \frac{2\beta^{\tau_N}}{1-\beta}L. \tag{19}$$

Proof is omitted due to lack of space.

The upper-bound provided in Theorem 2 requires knowledge on ($\tilde{f}$, $\tilde{\psi}^*$,$\mathcal{Z}$). However, according to (14), $\tau_N$ is always equal or greater than $N$ i.e. $N \leq \tau_N$. Hence, one can obtain a more conservative error-bound (larger upper-bound) than the error-bound (upper-bound) in Theorem 2 that does not require any knowledge on ($\tilde{f}$, $\tilde{\psi}^*$,$\mathcal{Z}$) as follows.

**Corollary 1** *The difference in performance between $\Delta$ and $\Delta_N$ is bounded as follows:*

$$|\tilde{J}(\tilde{\boldsymbol{\psi}}^*) - \tilde{J}_N(\tilde{\boldsymbol{\psi}}_N^*)| \leq \frac{2\beta^N}{1-\beta} L. \tag{20}$$

*3) Finite-state RL algorithm:* Let $\mathcal{T}$ be a generic (model-based or model-free) RL algorithm designed for finite-state MDPs with infinite horizon discounted cost. By a generic RL algorithm, we mean any algorithm which fits to the following framework. At each iteration $k \in \mathbb{N}$, $\mathcal{T}$ knows the state of system, selects one action, and observes an instantaneous cost and the next state. The strategy learned (generated) by $\mathcal{T}$ converges to an optimal strategy as $k \to \infty$.

Let $\mathcal{T}$ operate on MDP $\Delta_N$ such that, at iteration $k$, it knows the state of the system $s_k \in \mathcal{S}_N$, selects one action $\boldsymbol{\gamma}_k \in \mathcal{G}$, and observes an instantaneous cost $\ell_k$ (which is a realization of the incurred cost $\ell(X_k, \mathbf{U}_k)$ at the original decentralized system). According to (11) and (15), we have

$$\mathbb{E}[\ell(X_k, \mathbf{U}_k)|S_{1:k}, \boldsymbol{\Gamma}_{1:k}] = \tilde{\ell}(S_k, \boldsymbol{\Gamma}_k), \quad S_k \in \mathcal{S}_N. \tag{21}$$

Hence, the instantaneous cost $\ell_k$ may be interpreted as a realization of the per-step cost of $\Delta_N$. Given dynamics $\tilde{f}$, $\mathcal{T}$ observes $z_k \in \mathcal{Z}$ and computes the next state $s_{k+1} = \tilde{f}(s_k, \boldsymbol{\gamma}_k, z_k)$. If $s_{k+1} \in \mathcal{S}_{N+1} \backslash \mathcal{S}_N$, then an action (or a sequence of actions) that transmits the state of system to a known state in $S_N$ i.e. $s_{k+1} = d^* \in D^*$ will be taken; otherwise, the system will continue from $s_{k+1} \in \mathcal{S}_N$.

Let $\tilde{\boldsymbol{\psi}}_N^k : \mathcal{S}_N \to \mathcal{G}$ be the learned strategy associated with RL algorithm $\mathcal{T}$ operating on MDP $\Delta_N$ at iteration $k$. Then, $\mathcal{T}$ updates its strategy $\tilde{\boldsymbol{\psi}}_N^{k+1}$ based on the observed cost $\ell_k$ and the transmitted next state $s_{k+1}$ by executing action $\boldsymbol{\gamma}_k$ at state $s_k$. We assume $\mathcal{T}$ converges to an optimal strategy $\tilde{\boldsymbol{\psi}}_N^*$ as $k \to \infty$ such that

$$\lim_{k \to \infty} |\tilde{J}_N(\tilde{\boldsymbol{\psi}}_N^k) - \tilde{J}_N(\tilde{\boldsymbol{\psi}}_N^*)| = 0. \tag{22}$$

Now, we need to convert (translate) the strategies in $\Delta_N$ to strategies in the original decentralized control problem described in Section 1.3, where the actual learning happens. Hence, we define a strategy $\boldsymbol{g}_N^k := (g_N^{k,i}, \dots, g_N^{k,n})$, at iteration $k$, as follows:

$$g_N^{k,i}(s, m^i) := \tilde{\boldsymbol{\psi}}_N^{k,i}(s)(m^i), \forall s \in \mathcal{S}_N, \forall m^i \in \mathcal{M}^i, \forall i, \tag{23}$$

where $\tilde{\boldsymbol{\psi}}_N^{k,i}$ denotes the $i$th term of $\tilde{\boldsymbol{\psi}}_N^k$.

**Theorem 3** *Let $J^*$ be the optimal performance of the original decentralized control system given in* (4)*. Then, the approximation error associated with using the learned strategy is bounded as follows:*

$$\lim_{k \to \infty} |J^* - J(\boldsymbol{g}_N^k)| = |\tilde{J}(\tilde{\boldsymbol{\psi}}^*) - \tilde{J}_N(\tilde{\boldsymbol{\psi}}_N^*)| \leq \epsilon_N, \tag{24}$$

*where $\epsilon_N = \frac{2\beta^{\tau_N}}{1-\beta} L \leq \frac{2\beta^N}{1-\beta} L$. Note that the error goes to zero exponentially in $N$.*

The proof follows from Theorem (1), Lemma 2, Theorem 2, and Corollary 1.

**Remark 2** *In general, a finite-state RL algorithm similar to Algorithm 1 can be derived for centralized POMDPs since we do not impose any restriction on the POMDP in step 2. The only required assumption is the existence of an action (or a sequence of actions) that prevents the system to transmit to information states that are generated after a sufficiently long time. The existence of such a reset strategy ("reset button") or an approximate reset strategy ("homing strategy") is a standard assumption in the literature. See [22], [23] and references therein.*

## 4. Decentralized Implementation

All agents are provided with state space $S_N$, action space $\mathcal{G}$, and dynamics $\tilde{f}$ as described in Section 3.2.1. Note that to obtain above knowledge, every agent must only know the information structure of system, action spaces $\{\mathcal{U}^i\}_{i=1}^n$, observation spaces $\{\mathcal{Y}^i\}_{i=1}^n$, discount factor $\beta$, upper-bound $L$ on per-step cost, and $\epsilon > 0$. In addition, agents may have partial knowledge of the model of system or may not.

Agents observe the instantaneous cost of the system. They have access to a common shared random number generator for the purpose of exploring the system consistently. Given state space $\mathcal{S}_N$, action space $\mathcal{G}$, and dynamics $\tilde{f}$, Algorithm 1 can be executed in a distributed manner because every agent can independently run Algorithm 1; agreeing upon a deterministic rule to break ties while using argmin ensures that all agents compute the same optimal strategy. Note that no more information needs to be shared; hence, *no communication* is required. According to Remark 1, Algorithm 1 also works for the pure decentralized control systems, when there is no information commonly shared between agents.

---

**Algorithm 1** Finite-State RL Algorithm

1: Given $\epsilon > 0$, choose a sufficiently large $N \in \mathbb{N}$ such that $\frac{2\beta^N}{1-\beta} L \leq \epsilon$. Then, construct state space $\mathcal{S}_N$, action space $\mathcal{G}$, and dynamics $\tilde{f}$. Initialize $s_1 = s^*$.

2: At iteration $k \in \mathbb{N}$, RL algorithm $\mathcal{T}$ picks $\boldsymbol{\gamma}_k = (\gamma_k^1, \dots, \gamma_k^n) \in \mathcal{G}$ at state $s_k \in \mathcal{S}_N$. Then, agent $i \in \{1, \dots, n\}$ takes action $u_k^i$ according to the chosen prescription $\gamma_k^i$ and local information $m_k^i \in \mathcal{M}^i$ as follows:

$$u_k^i = \gamma_k^i(m_k^i), \quad \forall i.$$

3: Based on the taken actions, the system incurs a cost $\ell_k$, evolves, and generates new information i.e. $(\{m_{k+1}^i\}_{i=1}^n, z_k)$. Every agent $i$ observes $z_k$ because it is common observation. Based on $z_k$, all agents consistently compute the next state

$$s_{k+1} = \tilde{f}(s_k, \boldsymbol{\gamma}_k, z_k).$$

If $s_{k+1} \notin \mathcal{S}_N$, then agents take an action (or a sequence of actions) that transmits the state of system to a state $s_{k+1} = d^* \in \mathcal{S}_N$; otherwise, the system proceeds from $s_{k+1} \in \mathcal{S}_N$. Note that during the reset process, the algorithm is paused till the system lands in a state in $S_N$.

4: $\mathcal{T}$ updates its strategy from $\tilde{\boldsymbol{\psi}}_N^k$ to $\tilde{\boldsymbol{\psi}}_N^{k+1}$ based on performing action $\boldsymbol{\gamma}_k$ at state $s_k$ and transmission to next state $s_{k+1}$ with instantaneous cost $\ell_k$.

5: $k \leftarrow k + 1$, and go to step 2 until termination.

---

Suppose that the generic RL algorithm $\mathcal{T}$ in Section 3.2.3 is Q-learning. Then, in off-line learning, every agent is allowed to have different step sizes (independently from the step sizes of other agents). However, in on-line learning, since we also need to consistently exploit the system, the step sizes should be chosen consistently, e.g., based on the number of visit to pair of state and prescription, i.e., $(s, \boldsymbol{\gamma})$.

## 5. Example: MABC

In this section, we provide an example to illustrate our approach. In this example, we consider the setup of partial knowledge of the model.

### A. Problem Formulation

Consider a two-user multiaccess broadcast system. At time $t$, $W_t^i \in \{0,1\}$ packets arrive at each user according to independent Bernoulli processes with $\mathbb{P}(W_t^i = 1) = p^i \in (0,1)$, $i = 1,2$. Each user may store only $X_t^i \in \{0,1\}$ packets in a buffer. If a packet arrives when the user-buffer is full, the packet is dropped. Both users may transmit $U_t^i \in \{0,1\}$ packets over a shared broadcast medium. A user can transmit only if it has a packet, thus $U_t^i \leq X_t^i$. If only one user transmits at a time, the transmission is successful and the transmitted packet is removed from the queue. If both users transmit simultaneously, packets "collide" and remain in the queue. Thus, the state update for users 1 and 2 is:

$$X_{t+1}^i = \min(X_t^i - U_t^i + U_t^1 U_t^2 + W_t^i, 1), \quad i = 1,2 \quad (25)$$

Due to the broadcast nature of the communication channel, each user observes the transmission decision of the other user i.e. information at each user at time $t$ is $(X_t^i, \mathbf{U}_{1:t-1})$, $i \in \{1,2\}$. Each user chooses a transmission decision as

$$U_t^i = g_t^i(X_t^i, \mathbf{U}_{1:t-1}), \quad i = 1,2, \quad (26)$$

where only actions $U_t^i \leq X_t^i$ are feasible. Similar to Section 1.3, we denote the *control strategy* by $\boldsymbol{g} = (\boldsymbol{g}^1, \boldsymbol{g}^2)$. The per unit cost $\ell(u_t^1, u_t^2)$ is defined to reflect the quality of transmission at time $t$ as follows:

$$\ell(\mathbf{x}_t, \mathbf{u}_t) = \begin{cases} 0 & u_t^1 = 0, u_t^2 = 0 \\ \ell^1 \leq 0 & u_t^1 = 1, u_t^2 = 0 \\ \ell^2 \leq 0 & u_t^1 = 0, u_t^2 = 1 \\ \ell^3 & u_t^1 = 1, u_t^2 = 1 \end{cases} \quad (27)$$

where $|\ell^j| \leq L, j = 1,2,3$. The performance of strategy $\boldsymbol{g}$ is measured by

$$J(\boldsymbol{g}) = \mathbb{E}^{\boldsymbol{g}}\Big[\sum_{t=1}^{\infty} \beta^{t-1} \ell(\mathbf{X}_t, \mathbf{U}_t)\Big]. \quad (28)$$

where $\beta \in (0,1)$. The case of symmetric arrivals ($p^1 = p^2$) was considered in [4], [5]. In recent years, the above model has been used as a benchmark for decentralized stochastic control problems [7], [8], [9]. We are interested in the following problem.

**Problem 2** *Given any $\epsilon > 0$, without knowing the arrival probabilities $p^1$ and $p^2$, and cost functions $\ell^1, \ell^2, \ell^3$, develop a decentralized Q-learning algorithm for both users such that users consistently learn an $\epsilon$-optimal strategy $\boldsymbol{g}^*$.*

### B. Decentralized Q-learning Algorithm

In this section, we follow the proposed two-step approach to develop a finite-state RL algorithm.

*1) An Equivalent Centralized POMDP:* In this step, we follow [6] and obtain the equivalent centralized POMDP for the completely known model as described in Section 3.1.

The common information shared between users is $C_t = \mathbf{U}_{1:t-1}$. Define $Z_t = C_{t+1} \backslash C_t = \mathbf{U}_t$. At time $t$, the coordinator observes $C_t = Z_{1:t-1}$ and prescribes $\gamma_t^i \colon X_t^i \mapsto U_t^i$ that tell each agent how to use their local information to generate the control action. For this specific model, the prescription $\gamma^i$ is completely specified by $A_t^i := \gamma_t^i(1)$ (since $\gamma_t^i(0)$ is always 0). Hence,

$$U_t^i = \gamma_t^i(X_t^i) = A_t^i \cdot X_t^i \quad (29)$$

Therefore, we may equivalently assume that the coordinator generates actions $\mathbf{A}_t = (A_t^1, A_t^2)$. The agents are passive and generate actions $(U_t^1, U_t^2)$ according to (29). Hence, at time $t$, the coordinator prescribes action $\mathbf{A}_t \in \{0,1\}^2$ and observes $Z_t = \mathbf{U}_t \in \{0,1\}^2$.

Following [6], define $\mathbf{\Pi}_t = (\Pi_t^1, \Pi_t^2)$, $\Pi_t^i = \mathbb{P}(X_t^i = 1 \mid \mathbf{U}_{1:t-1}, \mathbf{A}_{1:t-1})$, as information state for the coordinated system with initial state $\mathbf{\Pi}_1 = (p^1, p^2)$. It is shown in [6]:

1) The information state $\mathbf{\Pi}_t$ evolves according to

$$\mathbf{\Pi}_{t+1} = \phi(\mathbf{\Pi}_t, \mathbf{A}_t, \mathbf{U}_t) \quad (30)$$

where

$$\phi(\mathbf{\Pi}_t,\mathbf{A}_t,\mathbf{U}_t) = \begin{cases} (T_1\Pi_t^1, T_2\Pi_t^2) & \mathbf{A}_t = (0,0) \\ (p^1, T_2\Pi_t^2) & \mathbf{A}_t = (1,0) \\ (T_1\Pi_t^1, p^2) & \mathbf{A}_t = (0,1) \\ (1,1) & \mathbf{A}_t = (1,1), \mathbf{U}_t = (1,1) \\ (p^1, p^2) & \mathbf{A}_t = (1,1), \mathbf{U}_t \neq (1,1) \end{cases} \quad (31)$$

where $(p^1, p^2)$ are arrival rates and operator $T_i$ is given by $T_i q = (1-p^i)(1-q), \quad i = 1,2$.

2) The expected cost function is as follows:

$$\hat{\ell}(\mathbf{\Pi}_t,\mathbf{A}_t) = \begin{cases} 0, & \mathbf{A}_t = (0,0) \\ \ell^1\Pi_t^1, & \mathbf{A}_t = (1,0) \\ \ell^2\Pi_t^2, & \mathbf{A}_t = (0,1) \\ \ell^1\Pi_t^1 + \ell^2\Pi_t^2 + (\ell^3 - \ell^1 - \ell^2)\Pi_t^1\Pi_t^2 & \mathbf{A}_t = (1,1) \end{cases} \quad (32)$$

The action $(0,0)$ that corresponds to not transmitting is dominated by the actions $(1,0)$ or $(0,1)$. Therefore, with no loss of optimality, action $(0,0)$ is removed. In the sequel, we denote $\mathcal{A} := \{(0,1), (1,0), (1,1)\}$ as the action space of the coordinator.

We denote $\mathcal{R}$ as the reachable set of above centralized POMDP that contains all the realizations of $\boldsymbol{\pi}_t$ generated by $\boldsymbol{\pi}_{t+1} = \phi(\boldsymbol{\pi}_t, \boldsymbol{a}, \boldsymbol{u}), \forall \boldsymbol{a} \in \mathcal{A}, \forall \mathbf{u} \in \{0,1\}^2, \forall t \in \mathbb{N}$, with initial information state $\boldsymbol{\pi}_1 = (p^1, p^2)$. Thus, the reachable set $\mathcal{R}$ is given by

$$\begin{aligned} \mathscr{R} := &\{(1,1), (1,p^1), (p^2,1), (p^1,p^2)\} \\ &\cup \{(p^1, T_2^n p^2) : n \in \mathbb{N}\} \cup \{(T_1^n p^1, p^2) : n \in \mathbb{N}\}, \end{aligned} \quad (33)$$

where $T_i^n q = T_i(T_i^{n-1} q)$. According to Theorem 1, we have

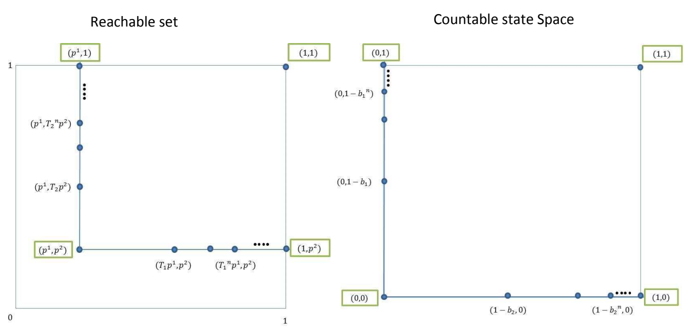


Fig. 1. It shows the reachable set $\mathcal{R}$ and the countable state space $\mathcal{S}$.

**Theorem 4** *Let $\boldsymbol{\psi}^*(\boldsymbol{\pi})$ be any argmin of the right-hand side of the following dynamic program. For $\boldsymbol{\pi} \in \mathcal{R}$,*

$$V(\boldsymbol{\pi}) = \min_{\boldsymbol{a}} (\hat{\ell}(\boldsymbol{\pi},\boldsymbol{a}) + \beta \mathbb{E}[V(\phi(\boldsymbol{\pi},\boldsymbol{a},\mathbf{U}_t)) | \boldsymbol{\Pi}_t = \boldsymbol{\pi}, \boldsymbol{A}_t = \boldsymbol{a}])$$

*where $\boldsymbol{a} \in \mathcal{A}$. The stationary strategy $\boldsymbol{g}^* = (g^{1,*}, g^{2,*})$ is optimal such that*

$$g^{i,*}(\boldsymbol{\pi}, x) = \boldsymbol{\psi}^{i,*}(\boldsymbol{\pi}) \cdot x, \quad \forall \boldsymbol{\pi} \in \mathcal{R}, x \in \{0,1\}, i = 1,2$$

*where $\boldsymbol{\psi}^{i,*}$ denotes ith term of $\boldsymbol{\psi}^*$.*

*2) Q-learning algorithm for the POMDP:* Let $b_1, b_2$ be any arbitrary number in $(0,1)$ and $B : \mathcal{R} \mapsto \mathbb{Z}^{+2}$ be a bijective function that maps each state of $\mathscr{R}$ to a point in $\mathbb{Z}^{+2}$ as follows:

$$(0, 1 - b_1^n) = B(p^1, T_2^n p^2), (1 - b_2^n, 0) = B(T_1^n p^1, p^2), n \in \mathbb{N}$$

$$(0,1) = B(p^1,1), (1,0) = B(1,p^2), (1,1) = B(1,1), (0,0) = B(p^1,p^2)$$

where $\lim_{n\to\infty} B(p^1, T_2^n p^2) = B(p^1, 1)$ and $\lim_{n\to\infty} B(T_1^n p^1, p^2) = B(1, p^2)$. Define a countable-state MDP $\Delta$ with state space $\mathcal{S}$, action space $\mathcal{A}$, dynamics $\tilde{f}$, and cost function $\tilde{\ell}$ as follows:

**(F1)** Let $\mathcal{S} = \{S_k\}_{k=1}^{\infty}$ be the state space, where $S_1 = \{(0,0)\}$ and $S_k = \{(0,0),(0,1),(1,0),(1,1),(0,1 - b_1^i),(1 - b_2^i, 0)\}_{i=1}^{k-1}$, $k \geq 2$. The action space is $\mathcal{A} = \{(0,1),(1,0),(1,1)\}$. The initial state $S_1 = (0,0)$. The state $S_t \in \mathcal{S}_k$, $k \leq t$, evolves as follows: for $\mathbf{A}_t \in \mathcal{A}, \mathbf{U}_t \in \{0,1\}^2$,

$$S_{t+1} = \tilde{f}(S_t, \mathbf{A}_t, \mathbf{U}_t), \quad S_{t+1} \in \mathcal{S}_{k+1}. \tag{34}$$

For ease of exposition of dynamics $\tilde{f}$, we denote every state $S_t \in \mathcal{S}_k$ in a format of $(1 - b_2^{k_2}, 1 - b_1^{k_1})$, where $k_1, k_2$ take value in the set of $\{0, 1, \dots, \infty\}$. Thus,

$$\tilde{f}(S_t, \mathbf{A}_t, \mathbf{U}_t) = \begin{cases} (0, 1 - b_1^{(k_1+1)}) & \mathbf{A}_t = (1,0) \\ (1 - b_2^{(k_2+1)}, 0) & \mathbf{A}_t = (0,1) \\ (1,1) & \mathbf{A}_t = (1,1), \mathbf{U}_t = (1,1) \\ (0,0) & \mathbf{A}_t = (1,1), \mathbf{U}_t \neq (1,1) \end{cases}$$

At time $t$, there is a cost given by

$$\tilde{\ell}(S_t, \mathbf{A}_t) = \hat{\ell}(B^{-1}(S_t), \mathbf{A}_t). \tag{35}$$

It is trivial to see that the tuple $\langle \{\mathcal{S}_k\}_{k=1}^{\infty}, B^{-1}, \tilde{f} \rangle$ is an IER because of the fact that

$$\phi(\cdot, \mathbf{a}, \mathbf{u}) = B^{-1}\left(\tilde{f}\,(B(\cdot), \mathbf{a}, \mathbf{u})\right), \quad \forall \mathbf{a} \in \mathcal{A}, \mathbf{u} \in \{0,1\}^2.$$

**Algorithm 2** Decentralized Q-learning Algorithm

1: Given $\epsilon > 0$, choose a sufficiently large $N \in \mathbb{N}$ such that $\frac{2\beta^N}{1-\beta} L \leq \epsilon$. Then, construct state space $\mathcal{S}_N$, action space $\mathcal{A}$, and dynamics $\tilde{f}$. Let $s_1 = (0,0)$. Initialize Q-functions with zero and step-sizes $\alpha$ with one i.e $Q(s,\mathbf{a}) = 0, \alpha(s,\mathbf{a}) = 1, \forall s \in \mathcal{S}_N, \forall \mathbf{a} \in \mathcal{A}$.

2: At iteration $k \in \mathbb{N}$, users uniformly pick a random action $\mathbf{a}_k \in \mathcal{A}$ at state $s_k \in \mathcal{S}_N$ by means of a common shared random number generator. Then, user $i \in \{1,2\}$ takes action $u_k^i$ according to the chosen $a_k^i$ and local information $x_k^i \in \{0,1\}$ as follows:
$$u_k^i = a_k^i \cdot x_k^i. \quad i = 1,2.$$

3: Based on the taken actions, the system incurs a cost $\ell_k$ and generates $(x_{k+1}^1, x_{k+1}^2, \mathbf{u}_k = (u_k^1, u_k^2))$. Since $\mathbf{u}_k$ is observable to both users, they consistently compute the next state
$$s_{k+1} = \tilde{f}(s_k, \mathbf{a}_k, \mathbf{u}_k).$$
If $s_{k+1} \notin \mathcal{S}_N$, user 1 transmits first and then, user 2 transmits, and the state of system will be transmitted to $s_{k+1} = (1 - b_2, 0)$; otherwise, the system proceeds from $s_{k+1} \in \mathcal{S}_N$.

4: Users update the corresponding Q-function associated with the pair $(s_k, \mathbf{a}_k)$ as follows:
$$Q(s_k,\mathbf{a}_k) \leftarrow (1 - \alpha(s_k,\mathbf{a}_k)) Q(s_k,\mathbf{a}_k) + \alpha(s_k,\mathbf{a}_k) \Big(\ell_k + \beta \min_{v \in \mathcal{A}} Q(s_{k+1}, v)\Big)$$
Also, the corresponding step-size are updated:
$$\frac{1}{\alpha(s_k, \mathbf{a}_k)} \leftarrow \frac{1}{\alpha(s_k, \mathbf{a}_k)} + 1.$$

5: $k \leftarrow k + 1$, and go to step 2 until termination.

**(F2)** State space $\mathcal{S}$, action space $\mathcal{A}$, and dynamics $\tilde{f}$ do not depend on the unknowns i.e. $(p^1, p^2, \ell^1, \ell^2, \ell^3)$.

The performance of a stationary strategy $\tilde{\boldsymbol{\psi}} : \mathcal{S} \mapsto \mathcal{A}$ is quantified by (16). According to Lemma 2, we can restrict attention in solving MDP $\Delta$ instead of the POMDP without loss of optimality. Let $\Delta_N$ be a finite-state MDP with state space $\mathcal{S}_N$ and action space $\mathcal{A}$. The initial state $S_1 = (0,0)$. At time $t$, state $S_t \in \mathcal{S}_N$ evolves as follows: for any $\mathbf{A}_t \in \mathcal{A}, \mathbf{U}_t \in \{0,1\}^2$,

$$S_{t+1} = \begin{cases} \tilde{f}(S_t, \mathbf{A}_t, \mathbf{U}_t) & \tilde{f}(S_t, \mathbf{A}_t, \mathbf{U}_t) \in \mathcal{S}_N \\ (1 - b_2, 0) & \tilde{f}(S_t, \mathbf{A}_t, \mathbf{U}_t) \in \mathcal{S}_{N+1} \backslash \mathcal{S}_N \end{cases} \tag{36}$$

In (36), whenever state $s_t$ steps out of $S_N$, the users take a sequence of actions as follows: At first, user 1 transmits and user 2 does not transmit, then user 2 transmits and user 1 does not transmit. This sequence of actions takes the system to state $(1 - b_2, 0) \in S_N, N \geq 2$. Now, we use standard Q-learning algorithm as the generic RL algorithm $\mathcal{T}$ to learn the optimal strategy of $\Delta_N$.

According to [3, Theorem 3], Q-functions in Algorithm 2 will converge to a $Q^*$ with probability one[1]. Let $Q^*$ be the resultant limit. Then, the optimal strategy $\tilde{\boldsymbol{\psi}}_N^*$ is as follows:

$$\tilde{\boldsymbol{\psi}}_N^* = \underset{\mathbf{a} \in \mathcal{A}}{\operatorname{argmin}} (Q^*(\cdot, \mathbf{a})). \tag{37}$$

The strategy $\boldsymbol{g}_N^*$ is $\epsilon_N$-optimal where

$$g_N^{i,*}(s)(x) := \tilde{\boldsymbol{\psi}}_N^{i,*}(s) \cdot x, \quad \forall s \in \mathcal{S}_N, x \in \{0,1\}, i = 1,2$$

where $\tilde{\boldsymbol{\psi}}_N^{i,*}$ denotes $i$th term of $\tilde{\boldsymbol{\psi}}_N^*$.

[1] In this example, every pair of (state,action) will be visited infinitely often by uniformly randomly picked actions.

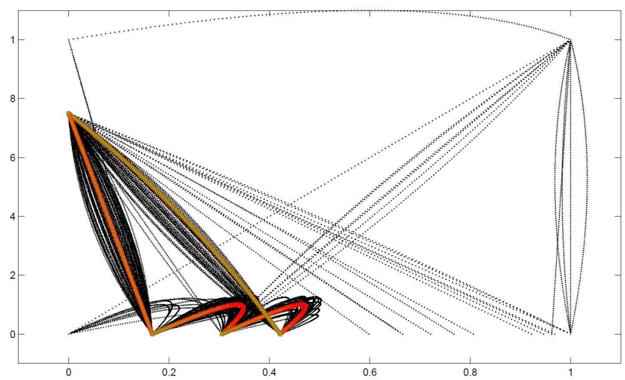

Fig. 2. This figure displays the learning procedure of optimal strategy in a few snapshots. It is seen that the state of the system is eventually trapped in the optimal recurrent class. The learning procedure is plotted in black and the optimal recurrent class is plotted in red. In this simulation, we use the following numerical values: $b_1 = 0.25, b_2 = 0.83, N = 20, \beta = 0.99, p^1 = 0.3, p^2 = 0.6, \ell^1 = \ell^2 = -1, \ell^3 = 0$.

*3) Numerical Results:* In this section, we provide a numerical simulation that shows the strategy learned by decentralized Q-learning Algorithm 2 converges to an optimal strategy when the arrival probabilities are $(p^1, p^2) = (0.3, 0.6)$ and the cost functions are $\ell^1 = \ell^2 = -1, \ell^3 = 0$.

Suppose users have no packets at the beginning. Users wait one time step to receive packets (i.e. user 1 receives a packet with $p^1 = 0.3$ probability and user 2 receives a packet with $p^2 = 0.6$ probability). At $t = 1$, action $(0, 1)$ is optimal i.e. the user 2 transmits and user 1 does not transmit. At $t \geq 2$, state $s_t$ enters a recurrent class under the optimal strategy, and stays there forever. The recurrent class includes four states: $(0, 1 - b_1^1)$, $(1 - b_2^1, 0)$, $(1 - b_2^2, 0)$, and $(1 - b_2^3, 0)$. One immediate result is that for any $N \geq 4$, state $S_t$ will never step out of $\mathcal{S}_N$ under the optimal strategy which implies $\tau_N = \infty$ and hence $\epsilon_N$ in Theorem 3 is zero (i.e. optimal strategy). For $t \geq 2$, the optimal strategy is a sequence of the following actions $(0, 1), (0, 1), (1, 0), (0, 1)$. Thus, it means that user 2 should transmit 3 times more than user 1 to minimize the number of collisions (maximize the number of successful transmission). Figure 2 displays a few snapshots of state $s_t$ governed by the strategy under the learning procedure where the learned strategy will eventually take state $s_t$ to the optimal recurrent class.

## 6. Conclusion

In this paper, we proposed a novel approach to develop a finite-state RL algorithm, for a large class of decentralized control systems with partial history sharing information structure, that guarantees $\epsilon-$team-optimal solution. We presented our approach in two steps. In the first step, we used the common information approach to obtain an equivalent centralized POMDP of the decentralized control problem. However, the resultant POMDP can not be used directly because it requires the complete knowledge of the model while the agents only know the model incompletely. Thus, in the second step, we introduced a new methodology to develop a RL algorithm for the obtained centralized POMDP. In particular, to remove the dependency of the complete knowledge, we introduced Incrementally Expanding Representation (IER) and based on that, we constructed a finite-state RL algorithm. In addition, we illustrated our approach by developing a decentralized Q-learning algorithm for two-user Multi Access Broadcast Channel (MABC), a benchmark example for decentralized control systems. The numerical simulations verify that the learned strategy converges to an optimal strategy.

## References


[1] S. Yuksel and T. Basar, Stochastic Networked Control Systems, 2013.

[2] A. Nayyar, A. Mahajan, and D. Teneketzis, Decentralized Stochastic Control with Partial History Sharing: A Common Information Approach, IEEE Trans. on Automatic Control, vol. 58, no. 7, Jul., 2013.

[3] J. N. Tsistsiklis, Synchronous Stochastic Approximation and Q-Learning, Machine Learning, vol. 16, pp. 185-202, 1994.

[4] M. G. Hluchyj and R. G. Gallager, Multiacces of a slotted channel by nitely many users, Proc. of Nat. Tel. Con., pp. 421-427, 1981.

[5] J. M. Ooi and G. W. Wornell, Decentralized control of a mutliple access broadcast channel: performance bounds, Proc. of the 35th Conference on Decision and Control, pp. 293-298, 1996.

[6] A. Mahajan, Optimal transmission policies for two-user multiple access broadcast using dynamic team theory, IEEE 48th Annual Allerton Conf. on, pp. 806–813, Sept., 2010.

[7] A. Mahajan, A. Nayyar, and D. Teneketzis, Identifying tractable decentralized control problems on the basis of information structure, Proc. 46th Annual Allerton Conf., pp. 1440-1449, 2008.

[8] S. Seuken and S. Zilberstein, Memory-bounded dynamic programming for DEC-POMDPs, in Proceedings of the 20th international joint conference on Artical intelligence, pp. 2009-2015, USA, 2007.

[9] J. S. Dibangoye, A.-I. Mouaddib, and B. Chaib-draa, Incremental pruning heuristic for solving DEC-POMDPs, Proc. of Workshop on Multiagent Sequential Decision Making in Uncertain Domains , 2008.

[10] M. T. J. Spaan, N. Vlassis, and F. C. A. Groen, High level coordination of agents based on multiagent Markov decision processes with roles, in Workshop on Coop. Rob., IEEE/RSJ Int., pp. 66-73, 2002.

[11] N. Vlassis, A concise introduction to multiagent systems and distributed AI,Univ. of Amsterdam, The Netherlands, Tech. Rep., 2003.

[12] J. R. Kok, M. T. J. Spaan, and N. Vlassis, Non-communicative multirobot coordination in dynamic environment, Robotics and Autonomous Systems, Vol. 50, No. 2-3, pp. 99-114, 2005.

[13] C. Claus and C. Boutilier, The dynamics of reinforcement learning in cooperative multiagent systems, 10th Conf. on Inn. Appl. of AI , pp. 746-752, 1998.

[14] S. Kapetanakis and D. Kudenko, Reinforcement learning of coordination in cooperative multi-agent systems, 14th conf. on Inn. Appl. of AI , pp. 326-331, 2002.

[15] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, Hysteretic Q-Learning : an algorithm for Decentralized Reinforcement Learning in Cooperative Multiagent Teams,

[16] J. huang, B. Yang, and D. Y. Liu, A distributed Q-learning Algorithm for Multi-Agent Team coordination, IEEE 40th Int. conf. on Mach. Ler. and Cyb., Vol. 1, pp. 108-113, Aug., 2005.

[17] A. Nayyar, A. Mahajan, and D. Teneketzis, Optimal control strategies in delayed sharing information structures, IEEE Tran. on Automatic Control, vol. 56, no. 7, pp. 1606-1620, Jul., 2011.

[18] J. M Ooi, Sh. M. Verbout, J. T. Ludwig, and G. W. Wornell, A separation theorem for periodic sharing information patterns in decentralized control, IEEE transactions on Automatic Control, vol. 42, no. 11, pp. 1546–1550, 1997.

[19] J. Arabneydi and A. Mahajan, Team optimal control of coupled subsystems with mean-field sharing, IEEE, Conference on decision and Control, Dec., 2014.

[20] A. Mahajan, Optimal Decentralized Control of Coupled agents with Control Sharing, IEEE Transaction on Automatic Control, vol. 58, no. 9, Sept., 2013.

[21] R. Murphey and P. Pardalos, Cooperative control and optimization, Springer Science & Business Media, vol. 66, 2002.

[22] K. Pivazyan and Y. Shoham, Polynomial-time reinforcement learning of near-optimal policies, AAAI/IAAI, pp 205–210, 2002.

[23] E. Even-Dar, Algorithms for Reinforcement Learning, Ph.D. thesis, TAV university, 2005.