Regret Bounds for Decentralized Learning in Cooperative Multi-Agent Dynamical Systems

Seyed Mohammad Asghari
University of Southern California
s.m.asghari.pari@gmail.com

Yi Ouyang
Preferred Networks America, Inc
ouyangyii@gmail.com

Ashutosh Nayyar
University of Southern California
ashutosn@usc.edu

Abstract

Regret analysis is challenging in Multi-Agent Reinforcement Learning (MARL) primarily due to the dynamical environments and the decentralized information among agents. We attempt to solve this challenge in the context of decentralized learning in multi-agent linear-quadratic (LQ) dynamical systems. We begin with a simple setup consisting of two agents and two dynamically decoupled stochastic linear systems, each system controlled by an agent. The systems are coupled through a quadratic cost function. When both systems' dynamics are unknown and there is no communication among the agents, we show that no learning policy can generate sub-linear in T regret, where T is the time horizon. When only one system's dynamics are unknown and there is one-directional communication from the agent controlling the unknown system to the other agent, we propose a MARL algorithm based on the construction of an auxiliary single-agent LQ problem. The auxiliary single-agent problem in the proposed MARL algorithm serves as an implicit coordination mechanism among the two learning agents. This allows the agents to achieve a regret within $O(\sqrt{T})$ of the regret of the auxiliary single-agent problem. Consequently, using existing results for single-agent LQ regret, our algorithm provides a $\tilde{O}(\sqrt{T})$ regret bound. (Here $\tilde{O}(\cdot)$ hides constants and logarithmic factors). Our numerical experiments indicate that this bound is matched in practice. From the two-agent problem, we extend our results to multi-agent LQ systems with certain communication patterns which appear in vehicle platoon control.

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

1 INTRODUCTION

Multi-agent systems arise in many different domains, including multi-player card games (Bard et al., 2019), robot teams (Stone and Veloso, 1998), vehicle formations (Fax and Murray, 2004), urban traffic control (De Oliveira and Camponogara, 2010), and power grid operations (Schneider et al., 1999). A multi-agent system consists of multiple autonomous agents operating in a common environment. Each agent gets observations from the environment (and possibly from some other agents) and, based on these observations, each agent chooses actions to collect rewards from the environment. The agents' actions may influence the environment dynamics and the reward of each agent. Multi-agent systems where the environment model is known to all agents have been considered under the frameworks of multi-agent planning (Oliehoek et al., 2016), decentralized optimal control (Yüksel and Başar, 2013), and non-cooperative game theory (Basar and Olsder, 1999). In realistic situations, however, the environment model is usually only partially known or even totally unknown. Multi-Agent Reinforcement Learning (MARL) aims to tackle the general situation of multi-agent sequential decision-making where the environment model is not completely known to the agents. In the absence of the environmental model, each agent needs to learn the environment while interacting with it to collect rewards. In this work, we focus on decentralized learning in a cooperative multi-agent setting where all agents share the same reward (or cost) function.

A number of successful learning algorithms have been developed for Single-Agent Reinforcement Learning (SARL) in single-agent environment models such as finite Markov decision processes (MDPs) and linear quadratic (LQ) dynamical systems. To extend SARL algorithms to cooperative MARL problems, one key challenge is the coordination among agents (Panait and Luke, 2005; Hernandez-Leal et al., 2017). In general, agents

have access to different information and hence agents may have different views about the environment from their different learning processes. This difference in perspectives makes it difficult for agents to coordinate their actions for maximizing rewards.

One popular method to resolve the coordination issue is to have a central entity collect information from all agents and determine the policies for each agent. Several works generalize SARL methods to multi-agent settings with such an approach by either assuming the existence of a central controller or by training a centralized agent with information from all agents in the learning process, which is the idea of *centralized training with decentralized execution* (Foerster et al., 2016; Dibangoye and Buffet, 2018; Hernandez-Leal et al., 2018). With centralized information, the learning problem reduces to a single-agent problem which can be readily solved by SARL algorithms. In many real-world scenarios, however, there does not exist a central controller or a centralized agent receiving all the information. Agents have to learn in a decentralized manner based on the observations they get from the environment and possibly from some other agents. In the absence of a centralized entity, an efficient MARL algorithm should guide each agent to learn the environment while maintaining certain level of coordination among agents.

Moreover, in online learning scenarios, the trade-off between exploration and exploitation is critical for the performance of a MARL algorithm during learning (Hernandez-Leal et al., 2017). Most existing SARL algorithms balance the exploration-exploitation trade off by controlling the posterior estimates/beliefs of the agent. Since multiple agents have decentralized information in MARL, it is not possible to directly extend SARL methods given the agents' distinct posterior estimates/beliefs. Furthermore, the fact that each agent's estimates/beliefs may be private to itself prevents any direct imitation of SARL. These issues make it extremely challenging to design coordinated policies for multiple agents to learn the environment and maintain good performance during learning. In this work, we attempt to solve this challenge in online decentralized MARL in the context of multi-agent learning in linear-quadratic (LQ) dynamical systems. Learning in LQ systems is an ideal benchmark for studying MARL due to a combination of its theoretical tractability and its practical application in various engineering domains (Aström and Murray, 2010; Abbeel et al., 2007; Levine et al., 2016; Abeille et al., 2016; Lazic et al., 2018).

We begin with a simple setup consisting of two agents and two stochastic linear systems as shown in Figure 1. The systems are dynamically decoupled but coupled through a quadratic cost function. In spite of its simplicity, this setting illustrates some of the inherent challenges and potential results in MARL. When the parameters of both systems 1 and 2 are known to both agents, the optimal solution to this multi-agent control problem can be computed in closed form (Ouyang et al., 2018). We consider the settings where the system parameters are completely or partially unknown and formulate an online MARL problem to minimize the agents' regret during learning. The regret is defined to be the difference between the cost incurred by the learning agents and the steady-state cost of the optimal policy computed using complete knowledge of the system parameters.

We provide a finite-time regret analysis for a decentralized MARL problem with controlled dynamical systems. In particular, we show that

1. First, if all parameters of a system are unknown, then both agents should receive information about the state of this system; otherwise, there is **no** learning policy that can guarantee sub-linear regret for all instances of the decentralized MARL problem (Theorem 1 and Lemma 2).

2. Further, when only one system's dynamics are unknown and there is one-directional communication from the agent controlling the unknown system to the other agent, we propose a MARL algorithm with regret bounded by $\tilde{O}(\sqrt{T})$ (Theorem 2 and Corollary 1).

The proposed MARL algorithm builds on an auxiliary SARL problem constructed from the MARL problem. Each agent constructs the auxiliary SARL problem by itself and applies a SARL algorithm $\mathcal{A}$ to it. Each agent chooses its action by modifying the output of the SARL algorithm $\mathcal{A}$ based on its information at each time. In our proposed algorithm, the auxiliary SARL problem serves as the critical coordination tool for the two agents to learn individually while jointly maintaining an exploration-exploitation balance. In fact, we will later show that the SARL dynamics can be seen as the filtering equation for the common state estimate of the agents.

We show that the regret achieved by our MARL algorithm is upper bounded by the regret of the SARL algorithm $\mathcal{A}$ in the auxiliary SARL problem plus an overhead bounded by $O(\sqrt{T})$. This implies that the MARL regret can be bounded by $\tilde{O}(\sqrt{T})$ by letting $\mathcal{A}$ be one of the state-of-the-art SARL algorithms for LQ systems which achieve $\tilde{O}(\sqrt{T})$ regret (Abbasi-Yadkori and Szepesvári, 2011; Faradonbeh et al., 2017, 2019). Our numerical experiments indicate that this bound is matched in simulations. From the two-agent problem, we extend our results

to multi-agent LQ systems with certain communication patterns which appear in vehicle platoon control.

Related work. There exists a rich and expanding body of work in the field of MARL (Littman, 1994; Nowé et al., 2012). Despite recent successes in empirical works including the adaptation of deep learning (Hernandez-Leal et al., 2018), many theoretical aspects of MARL are still under-explored. As multiple agents learn and adapt their policies, the environment is non-stationary from a single agent's perspective (Hernandez-Leal et al., 2017). Therefore, convergence guarantees of SARL algorithms are mostly invalid for MARL problems. Several works have extended SARL algorithms to independent or cooperative agents and analyzed their convergence properties (Tan, 1993; Greenwald et al., 2003; Kar et al., 2013; Amato and Oliehoek, 2015; Zhang et al., 2018; Gagrani and Nayyar, 2018; Wai et al., 2018). However, most of these works do not take into account the performance during learning except Bowling (2005). The algorithm of Bowling (2005) has a regret bound of $O(\sqrt{T})$, but the analysis is limited to repeated games. In contrast, we are interested in MARL in dynamical systems.

Regret analysis in online learning has been mostly focusing on multi-armed bandit (MAB) problems (Lai and Robbins, 1985). Upper-Confidence-Bound (UCB) (Auer and Fischer, 2002; Bubeck and Cesa-Bianchi, 2012; Dani et al., 2008) and Thompson Sampling (Thompson, 1933; Kaufmann et al., 2012; Agrawal and Goyal, 2013; Russo and Van Roy, 2014) are the two popular classes of algorithms that provide near-optimal regret guarantees in single-agent MAB. These ideas have been extended to certain multi-agent MAB settings (Liu and Zhao, 2010; Korda and Shuai, 2016; Nayyar and Jain, 2016). Multi-agent MAB can be viewed as a special class of MARL problems, but the lack of dynamics in MAB environments makes a drastic difference from the dynamical setting in this paper.

In the learning of dynamical systems, recent works have adopted concepts from MAB to analyze the regret of SARL algorithms in MDP (Jaksch et al., 2010; Osband et al., 2013; Gopalan and Mannor, 2015; Ouyang et al., 2017b) and LQ systems (Abbasi-Yadkori and Szepesvári, 2011; Faradonbeh et al., 2017, 2019; Ouyang et al., 2017a; Abbasi-Yadkori and Szepesvári, 2015; Abeille and Lazaric, 2018). Our MARL algorithm builds on these SARL algorithms by using the novel idea of constructing an auxiliary SARL problem for multi-agent coordination.

Notation. The collection of matrices A^1 and A^2 (resp. vectors x^1 and x^2) is denoted as $A^{1,2}$ (resp. $x^{1,2}$). Given column vectors x^1 and x^2, the notation $\mathbf{vec}(x^1, x^2)$ is used to denote the column vector formed by stacking x^1 on top of x^2. We use $[P^{\cdot,\cdot}]_{1,2}$ and $\mathbf{diag}(P^1, P^2)$ to denote the following block matrices, $[P^{\cdot,\cdot}]_{1,2} := \begin{bmatrix} P^{11} & P^{12} \\ P^{21} & P^{22} \end{bmatrix}$, $\mathbf{diag}(P^1, P^2) = \begin{bmatrix} P^1 & \mathbf{0} \\ \mathbf{0} & P^2 \end{bmatrix}$.

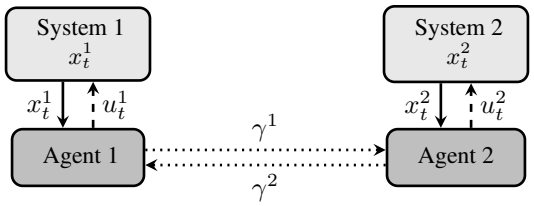


Figure 1: Two-agent system model. Solid lines indicate communication links, dashed lines indicate control links, and dotted lines indicate the possibility of information sharing.

2 PROBLEM FORMULATION

Consider a multi-agent Linear-Quadratic (LQ) system consisting of two systems and two associated agents as shown in Figure 1. The linear dynamics of systems 1 and 2 are given by

$$\begin{aligned} x_{t+1}^1 &= A_*^1 x_t^1 + B_*^1 u_t^1 + w_t^1, \\ x_{t+1}^2 &= A_*^2 x_t^2 + B_*^2 u_t^2 + w_t^2, \end{aligned} \tag{1}$$

where for $n \in \{1, 2\}$, $x_t^n \in \mathbb{R}^{d_x^n}$ is the state of system n and $u_t^n \in \mathbb{R}^{d_u^n}$ is the action of agent n. $A_*^{1,2}$ and $B_*^{1,2}$ are system matrices with appropriate dimensions. We assume that for $n \in \{1, 2\}$, w_t^n, $t \geq 0$, are i.i.d with standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The initial states $x_0^{1,2}$ are assumed to be fixed and known.

The overall system dynamics can be written as,

$$x_{t+1} = A_* x_t + B_* u_t + w_t, \tag{2}$$

where we have defined $x_t = \mathbf{vec}(x_t^1, x_t^2), u_t = \mathbf{vec}(u_t^1, u_t^2), w_t = \mathbf{vec}(w_t^1, w_t^2)$, $A_* = \mathbf{diag}(A_*^1, A_*^2)$, and $B_* = \mathbf{diag}(B_*^1, B_*^2)$.

At each time t, agent n, $n \in \{1, 2\}$, perfectly observes the state x_t^n of its respective system. The pattern of information sharing plays an important role in the analysis of multi-agent systems. In order to capture different information sharing patterns between the agents, let $\gamma^n \in \{0, 1\}$ be a fixed binary variable indicating the availability of a communication link from agent n to the other agent. Then, i_t^n which is the information sent by agent n to the other agent can be written as,

$$i_t^n = \begin{cases} x_t^n & \text{if } \gamma^n = 1 \\ \varnothing & \text{otherwise} \end{cases}. \tag{3}$$

At each time t, agent n's action is a function π_t^n of its information h_t^n, that is, $u_t^n = \pi_t^n(h_t^n)$ where $h_t^1 = \{x_{0:t}^1, u_{0:t-1}^1, i_{0:t}^2\}$ and $h_t^2 = \{x_{0:t}^2, u_{0:t-1}^2, i_{0:t}^1\}$. Let $\pi = (\pi^1, \pi^2)$ where $\pi^n = (\pi_0^n, \pi_1^n, \ldots)$. We will look at two following information sharing patterns:[1]

1. No information sharing ($\gamma^1 = \gamma^2 = 0$),
2. One-way information sharing from agent 1 to agent 2 ($\gamma^1 = 1, \gamma^2 = 0$).

At time t, the system incurs an instantaneous cost $c(x_t, u_t)$, which is a quadratic function given by

$$c(x_t, u_t) = x_t^\intercal Q x_t + u_t^\intercal R u_t, \tag{4}$$

where $Q = [Q^{\cdot,\cdot}]_{1,2}$ is a known symmetric positive semi-definite (PSD) matrix and $R = [R^{\cdot,\cdot}]_{1,2}$ is a known symmetric positive definite (PD) matrix.

2.1 THE OPTIMAL MULTI-AGENT LINEAR-QUADRATIC PROBLEM

Let $\theta_*^n = [A_*^n, B_*^n]$ be the dynamics parameter of system n, $n \in \{1, 2\}$. When θ_*^1 and θ_*^2 are perfectly known to the agents, minimizing the infinite horizon average cost is a multi-agent stochastic Linear Quadratic (LQ) control problem. Let $J(\theta_*^{1,2})$ be the optimal infinite horizon average cost under $\theta_*^{1,2}$, that is,

$$J(\theta_*^{1,2}) = \inf_{\pi} \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}^{\pi}[c(x_t, u_t) | \theta_*^{1,2}]. \tag{5}$$

We make the following standard assumption about the multi-agent stochastic LQ problem.

Assumption 1. *(A_*, B_*) is stabilizable[2] and $(A_*, Q^{1/2})$ is detectable[3].*

The above decentralized stochastic LQ problem has been studied by Ouyang et al. (2018). The following lemma summarizes this result.

Lemma 1 (Ouyang et al. (2018))**.** *Under Assumption 1, the optimal control strategies are given by*

$$u_t^1 = K^1(\theta_*^{1,2}) \begin{bmatrix} \hat{x}_t^1 \\ \hat{x}_t^2 \end{bmatrix} + \tilde{K}^1(\theta_*^1)(x_t^1 - \hat{x}_t^1),$$
$$u_t^2 = K^2(\theta_*^{1,2}) \begin{bmatrix} \hat{x}_t^1 \\ \hat{x}_t^2 \end{bmatrix} + \tilde{K}^2(\theta_*^2)(x_t^2 - \hat{x}_t^2), \tag{6}$$

where the gain matrices $K^1(\theta_*^{1,2}), K^2(\theta_*^{1,2}), \tilde{K}^1(\theta_*^1)$, and $\tilde{K}^2(\theta_*^2)$ can be computed offline[4] and $\hat{x}_t^n$, $n \in \{1, 2\}$, can be computed recursively according to

$$\hat{x}_0^n = x_0^n, \quad \hat{x}_{t+1}^n = \begin{cases} x_{t+1}^n & \text{if } \gamma^n = 1 \\ A_*^n \hat{x}_t^n + B_*^n K^n(\theta_*^{1,2})\, \mathbf{vec}(\hat{x}_t^1, \hat{x}_t^2) & \text{otherwise} \end{cases}. \tag{7}$$

2.2 THE MULTI-AGENT REINFORCEMENT LEARNING PROBLEM

The problem we are interested in is to minimize the infinite horizon average cost when the matrices A_* and B_* of the system are unknown. In this case, the control problem described by (1)-(4) can be seen as a Multi-Agent Reinforcement Learning (MARL) problem where both agents need to learn the system parameters $\theta_*^1 = [A_*^1, B_*^1]$ and $\theta_*^2 = [A_*^2, B_*^2]$ in order to minimize the infinite horizon average cost. The learning performance of policy π is measured by the cumulative regret over T steps defined as,

$$R(T, \pi) = \sum_{t=0}^{T-1} \left[c(x_t, u_t) - J(\theta_*^{1,2}) \right], \tag{8}$$

which is the difference between the performance of the agents under policy π and the optimal infinite horizon cost under full information about the system dynamics. Thus, the regret can be interpreted as a measure of the cost of not knowing the system dynamics.

3 AN AUXILIARY SINGLE-AGENT LQ PROBLEM

In this section, we construct an auxiliary single-agent LQ control problem based on the MARL problem of Section 2. This auxiliary single-agent LQ control problem is inspired by the *common information based coordinator* (which has been developed in non-learning settings in Nayyar et al. (2013) and Asghari et al. (2018) and the references therein). We will later use the auxiliary problem as a coordination mechanism for our MARL algorithm.

Consider a single-agent system with dynamics

$$x_{t+1}^\diamond = A_* x_t^\diamond + B_* u_t^\diamond + \begin{bmatrix} w_t^1 \\ \mathbf{0} \end{bmatrix}, \tag{9}$$

where $x_t^\diamond \in \mathbb{R}^{d_x^1 + d_x^2}$ is the state of the system, $u_t^\diamond \in \mathbb{R}^{d_u^1 + d_u^2}$ is the action of the auxiliary agent, w_t^1 is the noise vector of system 1 defined in (1), and matrices A_* and B_* are as defined in (2). The initial state $x_0^\diamond$ is

[1] The other possible pattern is two-way information sharing ($\gamma^1 = \gamma^2 = 1$). In this case, both agents observe the states of both systems. Due to the lack of space, we delegate this case to Appendix M.

[2] (A_*, B_*) is stabilizable if there exists a gain matrix K such that $A_* + B_* K$ is stable.

[3] $(A_*, Q^{1/2})$ is detectable if there exists a gain matrix H such that $A_* + HQ^{1/2}$ is stable.

[4] See Appendix J for the complete description of this result.

assumed to be equal to x_0. The action $u_t^\diamond = \pi_t^\diamond(h_t^\diamond)$ at time t is a function of the history of observations $h_t^\diamond = \{x_{0:t}^\diamond, u_{0:t-1}^\diamond\}$. The auxiliary agent's strategy is denoted by $\pi^\diamond = (\pi_1^\diamond, \pi_2^\diamond, \ldots)$. The instantaneous cost $c(x_t^\diamond, u_t^\diamond)$ of the system is a quadratic function given by

$$c(x_t^\diamond, u_t^\diamond) = (x_t^\diamond)^\intercal Q x_t^\diamond + (u_t^\diamond)^\intercal R u_t^\diamond, \qquad (10)$$

where matrices Q and R are as defined in (4).

When the parameters θ_*^1 and θ_*^2 are unknown, we will have a Single-Agent Reinforcement Learning (SARL) problem. In this problem, the regret of a policy $\pi^\diamond$ over T steps is given by

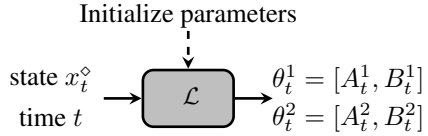
$$R^\diamond(T, \pi^\diamond) = \sum_{t=0}^{T-1} \left[c(x_t^\diamond, u_t^\diamond) - J^\diamond(\theta_*^{1,2}) \right], \qquad (11)$$

where $J^\diamond(\theta_*^{1,2})$ is the optimal infinite horizon average cost under $\theta_*^{1,2}$.

Existing algorithms for the SARL problem are generally based on the two following approaches: Optimism in the Face of Uncertainty (OFU) (Abbasi-Yadkori and Szepesvári, 2011; Faradonbeh et al., 2017, 2019) and Thompson Sampling (TS) (also known as posterior sampling) (Faradonbeh et al., 2017; Abbasi-Yadkori and Szepesvári, 2015; Abeille and Lazaric, 2018). In spite of the differences among these algorithms, all can be generally described as the AL-SARL algorithm (algorithm for the SARL problem). In this algorithm, at each time t, the agent interacts with a SARL learner (see Appendix I for a detailed description the SARL learner) by feeding time t and the state $x_t^\diamond$ to it and receiving *estimates* $\theta_t^1 = [A_t^1, B_t^1]$ and $\theta_t^2 = [A_t^2, B_t^2]$ of the unknown parameters $\theta_*^{1,2}$. Then, the agent uses $\theta_t^{1,2}$ to calculate the gain matrix $K(\theta_t^{1,2})$ (see Appendix J for a detailed description of this matrix) and executes the action $u_t^\diamond = K(\theta_t^{1,2}) x_t^\diamond$. As a result, a new state $x_{t+1}^\diamond$ is observed.

Algorithm 1 AL-SARL

Initialize $\mathcal{L}$ and $x_0^\diamond$
for $t = 0, 1, \ldots$ **do**
  Feed time t and state $x_t^\diamond$ to $\mathcal{L}$ and get θ_t^1 and θ_t^2
  Compute $K(\theta_t^{1,2})$
  Execute $u_t^\diamond = K(\theta_t^{1,2}) x_t^\diamond$
  Observe new state $x_{t+1}^\diamond$
end for



Among the existing algorithms, OFU-based algorithms of Abbasi-Yadkori and Szepesvári (2011); Faradonbeh et al. (2017, 2019) and the TS-based algorithm of Faradonbeh et al. (2017) achieve a $\tilde{O}(\sqrt{T})$ regret for the SARL problem (Here $\tilde{O}(\cdot)$ hides constants and logarithmic factors).

4 MAIN RESULTS

In this section, we start with the regret analysis for the case where the parameters of both systems are unknown (that is, θ_*^1 and θ_*^2 are unknown) and there is no information sharing between the agents (that is, $\gamma^1 = \gamma^2 = 0$). The detailed proofs for all results are in the appendix.

4.1 UNKNOWN θ_*^1 AND θ_*^2, NO INFORMATION SHARING ($\gamma^1 = \gamma^2 = 0$)

For the MARL problem of this section (it is called MARL1 for future reference), we show that there is no learning algorithm with a sub-linear in T regret for all instances of the MARL1 problem. The following theorem states this result.

Theorem 1. *There is no algorithm that can achieve a lower-bound better than $\Omega(T)$ on the regret of all instances of the MARL1 problem.*

A $\Omega(T)$ regret implies that the average performance of the learning algorithm has at least a constant gap from the ideal performance of informed agents. This prevent efficient learning performance even in the limit. Theorem 1 implies that in a MARL1 problem where the system dynamics are unknown, learning is not possible without communication between the agents. The proof of Theorem 1 also provides the following result.

Lemma 2. *Consider a MARL problem where the parameter of system 2 (that is, θ_*^2) is known to both agents and only the parameter of system 1 (that is, θ_*^1) is unknown. Further, there is no communication between the agents. Then, there is no algorithm that can achieve a lower-bound better than $\Omega(T)$ on the regret of all instances of this MARL problem.*

The above results imply that if the parameter of a system is unknown, both agents should receive information about this unknown system; otherwise, there is no learning policy π that can guarantee a sub-linear in T regret for all instances of this MARL problem.

In the next section, we assume that θ_*^2 is known to both agents and only θ_*^1 is unknown. Further, we assume the presence of a communication link from agent 1 to agent 2, that is, $\gamma^1 = 1$. This communication link allows agent 2 to receive feedback about the state x_t^1 of system 1 and

hence, remedies the impossibility of learning for agent 2.

4.2 UNKNOWN θ^1_*, ONE-WAY INFORMATION SHARING FROM AGENT 1 to AGENT 2 ($\gamma^1 = 1, \gamma^2 = 0$)

In this section, we consider the case where only system 1 is unknown and there is one-way communication from agent 1 to agent 2. Despite this one-way information sharing, the two agents still have different information. In particular, at each time agent 2 observes the state x^2_t of system 2 which is not available to agent 1. For the MARL of this section (it is called MARL2 for future reference), we propose the AL−MARL algorithm which builds on the auxiliary SARL problem of Section 3. AL−MARL algorithm is a decentralized multi-agent algorithm which is performed independently by the agents. Every agent independently constructs an auxiliary SARL problem where $x^\diamond_t = \mathbf{vec}(x^1_t, \check{x}^2_t)$ and applies an AL−SARL algorithm with its own learner $\mathcal{L}$ to it in order to learn the unknown parameter θ^1_* of system 1. In this algorithm, $\check{x}^2_t$ (described in the AL−MARL algorithm) is a proxy for $\hat{x}^2_t$ of (7) updated using the estimate θ^1_t instead of the unknown parameter θ^1_*.

At time t, each agent feeds $\mathbf{vec}(x^1_t, \check{x}^2_t)$ to its own SARL learner $\mathcal{L}$ and gets θ^1_t and θ^2_t. Note that both agents already know the true parameter θ^2_*, hence they only use θ^1_t to compute their gain matrix $K^{\texttt{agent\_ID}}(\theta^1_t, \theta^2_*)$ and use this gain matrix to compute their actions u^1_t and u^2_t according to the AL−MARL algorithm. Note that agent 2 needs $\tilde{K}^2(\theta^2_*)$ to calculate its actions u^2_t. However, we know that $\tilde{K}^2(\theta^2_*)$ is independent of the unknown parameter θ^1_* and hence, $\tilde{K}^2(\theta^2_*)$ can be calculated prior to the beginning of the algorithm. After the execution of the actions u^1_t and u^2_t by the agents, both agents observe the new state x^1_{t+1} and agent 2 further observes the new state x^2_{t+1}. Finally, each agent independently computes $\check{x}^2_{t+1}$.

Remark 1. *The state $x^\diamond_t$ of the auxiliary SARL can be interpreted as an estimate of the state x_t of the overall system* (2) *that each agent computes based on the common information between them. In fact, the SARL dynamics in* (9) *can be seen as the filtering equation for this common estimate.*

Remark 2. *We want to emphasize that unlike the idea of centralized training with decentralized execution (Foerster et al., 2016; Dibangoye and Buffet, 2018; Hernandez-Leal et al., 2018), the AL−MARL algorithm is an online decentralized learning algorithm. This means that there is no centralized learning phase in the setup where agents can collect information or have access to a simulator. The agents are simultaneously learning and controlling the system.*

Remark 3. *Since the SARL learner $\mathcal{L}$ can include taking samples and solving optimization problems, due to the independent execution of the AL−MARL algorithm, agents might receive different $\theta^{1,2}_t$ from their learner $\mathcal{L}$.*

Algorithm 2 AL−MARL

Input: agent_ID, learner $\mathcal{L}$, x^1_0, and x^2_0
Initialize $\mathcal{L}$ and $\check{x}^2_0 = x^2_0$
for $t = 0, 1, \ldots$ **do**
  Feed time t and state $\mathbf{vec}(x^1_t, \check{x}^2_t)$ to $\mathcal{L}$ and get $\theta^1_t = [A^1_t, B^1_t]$ and $\theta^2_t = [A^2_t, B^2_t]$
  Compute $K^{\texttt{agent\_ID}}(\theta^1_t, \theta^2_*)$
  if agent_ID $= 1$ **then**
    Execute $u^1_t = K^1(\theta^1_t, \theta^2_*)\, \mathbf{vec}(x^1_t, \check{x}^2_t)$
  else
    Execute $u^2_t = K^2(\theta^1_t, \theta^2_*)\, \mathbf{vec}(x^1_t, \check{x}^2_t) + \tilde{K}^2(\theta^2_*)(x^2_t - \check{x}^2_t)$
  end if
  Observe new state x^1_{t+1}
  Compute $\check{x}^2_{t+1} = A^2_* \check{x}^2_t + B^2_* K^2(\theta^1_t, \theta^2_*)\, \mathbf{vec}(x^1_t, \check{x}^2_t)$
  if agent_ID $= 2$ **then**
    Observe new state x^2_{t+1}
  end if
end for

In order to avoid the issue pointed out in Remark 3, we make an assumption about the output of the SARL learner $\mathcal{L}$.

Assumption 2. *Given the same time and same state input to the SARL learner $\mathcal{L}$, the outputs $\theta^{1,2}_t$ from different learners $\mathcal{L}$ are the same.*

Note that Assumption 2 can be easily achieved by setting the same initial sampling seed (if the SARL learner $\mathcal{L}$ includes taking samples) or by setting the same tie-breaking rule among possible similar solutions of an optimization problem (if the SARL learner $\mathcal{L}$ include solving optimization problems). Now, we present the following result which is based on Assumption 2.

Theorem 2. *Under Assumption 2, let $R(T, \texttt{AL−MARL})$ be the regret for the MARL2 problem under the policy of the AL−MARL algorithm and $R^\diamond(T, \texttt{AL−SARL})$ be the regret for the auxiliary SARL problem under the policy of the AL−SARL algorithm. Then for any $\delta \in (0, 1/e)$, with probability at least $1 - \delta$,*

$$R(T, \texttt{AL−MARL}) \leq R^\diamond(T, \texttt{AL−SARL}) + \log(\frac{1}{\delta})\tilde{K}\sqrt{T}.$$

This result shows that under the policy of the AL−MARL algorithm, the regret for the MARL2 problem is upper-bounded by the regret for the auxiliary SARL problem constructed in Section 3 under the policy of the AL−SARL algorithm plus a term bounded by $O(\sqrt{T})$.

Corollary 1. *AL−MARL algorithm with the OFU-based SARL learner $\mathcal{L}$ of Abbasi-Yadkori and Szepesvári (2011); Faradonbeh et al. (2017, 2019) or the TS-based SARL learner $\mathcal{L}$ of Faradonbeh et al. (2017) achieves a $\tilde{O}(\sqrt{T})$ regret for the MARL2 problem.*

Remark 4. *The idea of constructing a centralized problem for MARL is similar in spirit to the centralized algorithm perspective adopted in Dibangoye and Buffet (2018). However, we would like to emphasize that the auxiliary SARL problem is different from the centralized oMDP in Dibangoye and Buffet (2018). The oMDP is a **deterministic** MDP with no observations of the belief state. Our single agent problem is inspired by the common information based coordinator developed in non-learning settings in Nayyar et al. (2013) and Asghari et al. (2018). The difference from oMDP is reflected in the fact that the state evolution in the SARL is **stochastic** (see* (9)*).*

4.3 EXTENSION TO MARL PROBLEMS WITH MORE THAN 2 SYSTEMS AND 2 AGENTS

While the results of Sections 4.1 and 4.2 are for MARL problems with 2 systems and 2 agents, these results can be extended to MARL problems with an arbitrary number N of agents and systems in the following sense.

Lemma 3. *Consider a MARL problem with N agents and systems ($N \geq 2$). Suppose there is a system n and an agent m, $m \neq n$, such that system n is unknown and there is no communication from agent n to agent m. Then, there is no algorithm that can achieve a lower-bound better than $\Omega(T)$ on the regret of all instances of this MARL problem.*

The above lemma follows from the proof of Theorem 1.

Theorem 3. *Consider a MARL problem with N agents and systems ($N \geq 2$) where the first N_1 systems are unknown and the rest $N - N_1$ systems are known. Further, for any $1 \leq i \leq N_1$, there is communication from agent i to all other agents and for any $N_1 + 1 \leq j \leq N$, there is no communication from agent j to any other agent. Then, there is a learning algorithm that achieves a $\tilde{O}(\sqrt{T})$ regret for this MARL problem.*

The proof of above theorem requires constructing an auxiliary SARL problem and following the same steps as in the proof of Theorem 2.

Example 1. *Consider a platoon of N vehicles with one lead vehicle and $N-1$ followers. The objective of the platoon is to keep the distance between every two consecutive vehicles (the first vehicle is the lead vehicle) fixed. Each vehicle can adjust its velocity to achieve this goal. Assume that only the system dynamics of the lead vehicle are unknown but the position of this vehicle is available to all vehicles. If we define the position of the lead vehicle as the state of system 1 and the position of followers as the state of systems 2 to N, then this problem can be considered as an instance of our MARL problem. Note that since the location of a vehicle is independent of the location and velocity of other vehicles, in this example, the systems are decoupled.*

5 KEY STEPS IN THE PROOF OF THEOREM 2

STEP 1: SHOWING THE CONNECTION BETWEEN AUXILIARY SARL PROBLEM AND THE MARL2 PROBLEM

First, we present the following lemma that connects the optimal infinite horizon average cost $J^\diamond(\theta_*^{1,2})$ of the auxiliary SARL problem when $\theta_*^{1,2}$ are known (that is, the auxiliary single-agent LQ problem of Section 3) and the optimal infinite horizon average cost $J(\theta_*^{1,2})$ of the MARL2 problem when $\theta_*^{1,2}$ are known (that is, the multi-agent LQ problem of Section 2.1).

Lemma 4. $J(\theta_*^{1,2}) = J^\diamond(\theta_*^{1,2}) + \mathbf{tr}(D\Sigma)$, *where we have defined $D := Q^{22} + (\tilde{K}^2(\theta_*^2))^\intercal R^{22} \tilde{K}^2(\theta_*^2)$ and Σ is as defined in Lemma 10 in the appendix.*

Next, we provide the following lemma that shows the connection between the cost $c(x_t, u_t)$ in the MARL2 problem under the policy of the AL−MARL algorithm and the cost $c(x_t^\diamond, u_t^\diamond)$ in the auxiliary SARL problem under the policy of the AL−SARL algorithm.

Lemma 5. *At each time t, the following equality holds between the cost $c(x_t, u_t)$ in the MARL2 problem under the policy of the AL−MARL algorithm and the cost $c(x_t^\diamond, u_t^\diamond)$ in the auxiliary SARL problem under the policy of the AL−SARL algorithm,*

$$c(x_t, u_t)|_{\text{AL-MARL}} = c(x_t^\diamond, u_t^\diamond)|_{\text{AL-SARL}} + e_t^\intercal D e_t, \quad (12)$$

where $e_t = x_t^2 - \check{x}_t^2$ and D is as defined in Lemma 4.

STEP 2: USING THE SARL PROBLEM TO BOUND THE REGRET OF THE MARL2 PROBLEM

In this step, we use the connection between the auxiliary SARL problem and our MARL2 problem, which was established in Step 1, to prove Theorem 2. Note that from the definition of the regret in the MARL problem given by (8), we have,

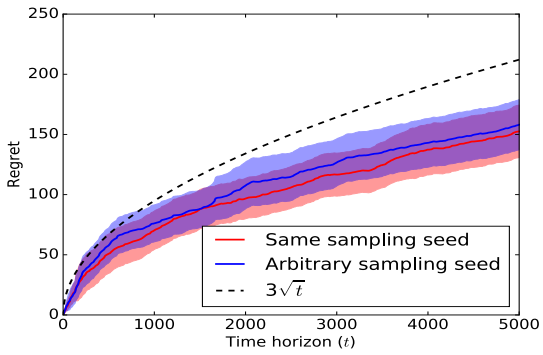


Figure 2: AL-MARL algorithm with the SARL learner of Faradonbeh et al. (2017)

$$R(T, \texttt{AL-MARL}) = \sum_{t=0}^{T-1} \left[c(x_t, u_t)|_{\texttt{AL-MARL}} - J(\theta_*^{1,2})\right]$$
$$= \sum_{t=0}^{T-1} \left[c(x_t^\diamond, u_t^\diamond)|_{\texttt{AL-SARL}} - J^\diamond(\theta_*^{1,2})\right]$$
$$+ \sum_{t=0}^{T-1} \left[e_t^\intercal D e_t - \mathbf{tr}(D\Sigma)\right]$$
$$\leq R^\diamond(T, \texttt{AL-SARL}) + \log(\frac{1}{\delta})\tilde{K}\sqrt{T}, \qquad (13)$$

where the second equality is correct because of Lemma 4 and Lemma 5. Further, the last inequality is correct because of the definition of the regret in the the SARL problem given by (11) and the fact that $\sum_{t=0}^{T-1} \left[e_t^\intercal D e_t - \mathbf{tr}(D\Sigma)\right]$ is bounded by $\log(\frac{1}{\delta})\tilde{K}\sqrt{T}$ from Lemma 11 in the appendix.

6 EXPERIMENTS

In this section, we illustrate the performance of the AL-MARL algorithm through numerical experiments. Our proposed algorithm requires a SARL learner. As the TS-based algorithm of Faradonbeh et al. (2017) achieves a $\tilde{O}(\sqrt{T})$ regret for a SARL problem, we use the SARL learner of this algorithm (The details for this SARL learner are presented in Appendix I).

We consider an instance of the MARL2 problem (See Appendix K for the details). The theoretical result of Theorem 2 holds when Assumption 2 is true. Since we use the TS-based learner of Faradonbeh et al. (2017), this assumption can be satisfied by setting the same sampling seed between the agents. Here, we consider both cases of same sampling seed and arbitrary sampling seed for the experiments. We ran 100 simulations and show the mean of regret with the 95% confidence interval for each scenario.

As it can be seen from Figure 2, for both of theses cases, our proposed algorithm with the TS-based learner $\mathcal{L}$ of Faradonbeh et al. (2017) achieves a $\tilde{O}(\sqrt{T})$ regret for our MARL2 problem, which matches the theoretical results of Corollary 1.

7 CONCLUSION

In this paper, we tackled the challenging problem of regret analysis in Multi-Agent Reinforcement Learning (MARL). We attempted to solve this challenge in the context of online decentralized learning in multi-agent linear-quadratic (LQ) dynamical systems. First, we showed that if a system is unknown, then all the agents should receive information about the state of this system; otherwise, there is no learning policy that can guarantee sub-linear regret for all instances of the decentralized MARL problem. Further, when a system is unknown but there is one-directional communication from the agent controlling the unknown system to the other agents, we proposed a MARL algorithm with regret bounded by $\tilde{O}(\sqrt{T})$.

The MARL algorithm is based on the construction of an auxiliary single-agent LQ problem. The auxiliary single-agent problem serves as an implicit coordination mechanism among the learning agents. The state of the auxiliary SARL can be interpreted as an estimate of the state of the overall system that each agent computes based on the common information among them. While there is a strong connection between the MARL and auxiliary SARL problems, the MARL problem is not reduced to a SARL problem. In particular, Lemma 5 shows that the costs of the two problems actually differ by a term that depends on the random process e_t, which is dynamically controlled by the MARL algorithm. Therefore, the auxiliary SARL problem is not equivalent to the MARL problem. Nevertheless, the proposed MARL algorithm can bound the additional regret due to the process e_t and achieve the same regret order as a SARL algorithm.

The use of the common state estimate plays a key role in the MARL algorithm. The current theoretical analysis uses this common state estimate along with some properties of LQ structure (e.g. certainty equivalence which connects estimates to optimal control (Kumar and Varaiya, 2015)) to quantify the regret bound. However, certainty equivalence is often used in general systems with continuous state and action spaces as a heuristic with some good empirical performance. This suggests that our algorithm combined with linear approximation of dynamics could potentially be applied to non-LQ systems as a heuristic. That is, each agent constructs an auxiliary SARL with the common estimate as the state, solves this SARL problem heuristically using approxi-

mate linear dynamics and/or certainty equivalence, and then modifies the SARL outputs according to the agent's private information.

Acknowledgements

This work was supported by NSF Grants ECCS 1509812 and ECCS 1750041 and ARO Award No. W911NF-17-1-0232.

References

Abbasi-Yadkori, Y. and Szepesvári, C. (2011). Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 1–26.

Abbasi-Yadkori, Y. and Szepesvári, C. (2015). Bayesian optimal control of smoothly parameterized systems. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 2–11. AUAI Press.

Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y. (2007). An application of reinforcement learning to aerobatic helicopter flight. In *Advances in neural information processing systems*, pages 1–8.

Abeille, M. and Lazaric, A. (2018). Improved regret bounds for thompson sampling in linear quadratic control problems. In *International Conference on Machine Learning*, pages 1–9.

Abeille, M., Serie, E., Lazaric, A., and Brokmann, X. (2016). LQG for portfolio optimization. Papers 1611.00997, arXiv.org.

Agrawal, S. and Goyal, N. (2013). Thompson sampling for contextual bandits with linear payoffs. In *ICML (3)*, pages 127–135.

Amato, C. and Oliehoek, F. A. (2015). Scalable planning and learning for multiagent pomdps. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Asghari, S. M., Ouyang, Y., and Nayyar, A. (2018). Optimal local and remote controllers with unreliable uplink channels. *IEEE Transactions on Automatic Control*, 64(5):1816–1831.

Aström, K. J. and Murray, R. M. (2010). *Feedback systems: an introduction for scientists and engineers*. Princeton university press.

Auer, Peter, N. C.-B. and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256.

Bard, N., Foerster, J. N., Chandar, S., Burch, N., Lanctot, M., Song, H. F., Parisotto, E., Dumoulin, V., Moitra, S., Hughes, E., et al. (2019). The hanabi challenge: A new frontier for ai research. *arXiv preprint arXiv:1902.00506*.

Basar, T. and Olsder, G. J. (1999). *Dynamic noncooperative game theory*, volume 23. Siam.

Bowling, M. (2005). Convergence and no-regret in multiagent learning. In *Advances in neural information processing systems*, pages 209–216.

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

Dani, V., Hayes, T. P., and Kakade, S. M. (2008). Stochastic linear optimization under bandit feedback. In *COLT*, pages 355–366.

De Oliveira, L. B. and Camponogara, E. (2010). Multiagent model predictive control of signaling split in urban traffic networks. *Transportation Research Part C: Emerging Technologies*, 18(1):120–139.

Dibangoye, J. S. and Buffet, O. (2018). Learning to act in decentralized partially observable mdps.

Faradonbeh, M. K. S., Tewari, A., and Michailidis, G. (2017). Optimism-based adaptive regulation of linear-quadratic systems. *arXiv preprint arXiv:1711.07230*.

Faradonbeh, M. K. S., Tewari, A., and Michailidis, G. (2019). On applications of bootstrap in continuous space reinforcement learning. *arXiv preprint arXiv:1903.05803*.

Fax, J. A. and Murray, R. M. (2004). Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465.

Foerster, J., Assael, I. A., de Freitas, N., and Whiteson, S. (2016). Learning to communicate with deep multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145.

Gagrani, M. and Nayyar, A. (2018). Thompson sampling for some decentralized control problems. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 1053–1058. IEEE.

Gopalan, A. and Mannor, S. (2015). Thompson sampling for learning parameterized markov decision processes. In *COLT*.

Greenwald, A., Hall, K., and Serrano, R. (2003). Correlated q-learning. In *ICML*, volume 3, pages 242–249.

Hernandez-Leal, P., Kaisers, M., Baarslag, T., and de Cote, E. M. (2017). A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183*.

Hernandez-Leal, P., Kartal, B., and Taylor, M. E. (2018). Is multiagent deep reinforcement learning the answer or the question? a brief survey. *arXiv preprint arXiv:1810.05587*.

Jaksch, T., Ortner, R., and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600.

Kar, S., Moura, J. M. F., and Poor, H. V. (2013). Qd-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus innovations. *IEEE Transactions on Signal Processing*, 61(7):1848–1862.

Kaufmann, E., Korda, N., and Munos, R. (2012). Thompson sampling: An asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*, pages 199–213. Springer.

Korda, Nathan, B. S. and Shuai, L. (2016). Distributed clustering of linear bandits in peer to peer networks. In *ICML*, pages 1301–1309.

Kumar, P. R. and Varaiya, P. (2015). *Stochastic systems: Estimation, identification, and adaptive control*, volume 75. SIAM.

Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22.

Lazic, N., Boutilier, C., Lu, T., Wong, E., Roy, B., Ryu, M., and Imwalle, G. (2018). Data center cooling using model-predictive control. In *Advances in Neural Information Processing Systems*, pages 3814–3823.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier.

Liu, K. and Zhao, Q. (2010). Distributed learning in multi-armed bandit with multiple players. *IEEE Transactions on Signal Processing*, 58(11):5667–5681.

Nayyar, A., Mahajan, A., and Teneketzis, D. (2013). Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Transactions on Automatic Control*, 58(7):1644–1658.

Nayyar, Naumaan, D. K. and Jain, R. (2016). On regret-optimal learning in decentralized multiplayer multi-armed bandits. *IEEE Transactions on Control of Network Systems*, 5(1):597–606.

Nowé, A., Vrancx, P., and De Hauwere, Y.-M. (2012). Game theory and multi-agent reinforcement learning. In *Reinforcement Learning*, pages 441–470. Springer.

Oliehoek, F. A., Amato, C., et al. (2016). *A concise introduction to decentralized POMDPs*, volume 1. Springer.

Osband, I., Russo, D., and Van Roy, B. (2013). (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pages 3003–3011.

Ouyang, Y., Asghari, S. M., and Nayyar, A. (2018). Optimal local and remote controllers with unreliable communication: the infinite horizon case. In *2018 Annual American Control Conference (ACC)*, pages 6634–6639. IEEE.

Ouyang, Y., Gagrani, M., and Jain, R. (2017a). Control of unknown linear systems with thompson sampling. In *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1198–1205. IEEE.

Ouyang, Y., Gagrani, M., Nayyar, A., and Jain, R. (2017b). Learning unknown markov decision processes: A thompson sampling approach. In *Advances in Neural Information Processing Systems*, pages 1333–1342.

Panait, L. and Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems*, 11(3):387–434.

Russo, D. and Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243.

Schneider, J., Wong, W. K., Moore, A., and Riedmiller, M. (1999). Distributed value functions. In *ICML*, pages 371–378.

Stone, P. and Veloso, M. (1998). Team-partitioned, opaque-transition reinforcement learning. *Robot Soccer World Cup*, pages 261–272.

Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pages 330–337.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294.

Wai, H.-T., Yang, Z., Wang, P. Z., and Hong, M. (2018). Multi-agent reinforcement learning via double averaging primal-dual optimization. In *Advances in Neural Information Processing Systems*, pages 9649–9660.

Yüksel, S. and Başar, T. (2013). *Stochastic networked control systems: Stabilization and optimization under information constraints*. Springer Science & Business Media.

Zhang, K., Yang, Z., Liu, H., Zhang, T., and Basar, T. (2018). Fully decentralized multi-agent reinforcement learning with networked agents. In *International Conference on Machine Learning*, pages 5867–5876.