Communication Efficient Parallel Reinforcement Learning

Mridul Agarwal[1] **Bhargav Ganguly**[2] **Vaneet Aggarwal**[2,1]

[1]School of Electrical and Computer Engineering., Purdue University, West Lafayette, Indiana, USA
[2]School of Industrial Engineering., Purdue University, West Lafayette, Indiana, USA

Abstract

We consider the problem where M agents interact with M identical and independent environments with S states and A actions using reinforcement learning for T rounds. The agents share their data with a central server to minimize their regret. We aim to find an algorithm that allows the agents to minimize the regret with infrequent communication rounds. We provide DIST-UCRL which runs at each agent and prove that the total cumulative regret of M agents is upper bounded as $\tilde{O}(DS\sqrt{MAT})$ for a Markov Decision Process with diameter D, number of states S, and number of actions A. The agents synchronize after their visitations to any state-action pair exceeds a certain threshold. Using this, we obtain a bound of $O\left(MSA\log(MT)\right)$ on the total number of communications rounds. Finally, we evaluate the algorithm against multiple environments and demonstrate that the proposed algorithm performs at par with an always communication version of the UCRL2 algorithm, while with significantly lower communication.

1 INTRODUCTION

Reinforcement Learning (RL) is being increasingly deployed and trained with parallel agents. In many cases, each agent interacts with identical and independent environments. For example, in autonomous cars, the agents do not share the environment as they are located possibly far away [Kiran et al., 2020]. Similarly, in ride/freight sharing services, different RL agents (vehicles) make decisions in parallel to minimize their costs and maximize the profits [Al-Abbasi et al., 2019, Manchella et al., 2021]. Further, consider an example of an e-commerce company using RL agents in servers for recommending products to customer. Based on the location of the customer, each customer's query may potentially be routed to a particular server (or agent) [Sankararaman et al., 2019]. Finally in the field of robotics, parallel robots are often deployed in practice [Hu et al., 2020, Sartoretti et al., 2019]. From these examples, we note that every agent would gain by sharing their data collected, otherwise they would be wasting the knowledge accumulated by the remaining agents. Such parallel policy learning with limited amount of communication is the focus of this paper.

Sharing of data across agents poses a new set of problems which comes from communicating the samples. If the agents communicate the observation tuple (state, action, reward, next state) at every time step, their communication complexity will increase. For various power constrained devices such as small robots, this luxury might not always be available [Sankararaman et al., 2019]. In this paper, we aim to work on the problem to reduce the number of communication steps while obtaining the same regret bounds as that of an strategy in which the agents always communicate.

For a system with M parallel agents, the parallel agents generate M times more data. From the lens of regret analysis, this setup loosely translates to a setup where a single agent works for time horizon of MT. For the setup where a single agent runs for MT steps, the regret is lower bounded by $O(\sqrt{DSMAT}$ and upper bounded by $\tilde{O}(DS\sqrt{MAT})$ [Jaksch et al., 2010, Agrawal and Jia, 2017]. We show that, if the agents take their actions sequentially and communicate after every interaction with their respective environments, then the agents can collectively obtain a regret bound of $\tilde{O}(DS\sqrt{MAT})$. This results in a faster convergence rate of $O(1/\sqrt{MT})$ to the optimal policy as compared to the convergence rate of $O(1/\sqrt{T})$ when using one RL agent. Note that in practice, a sequential decision making for parallel independent agents cannot be guaranteed. Thus, this result acts as a lower bound to parallel reinforcement learning. In this paper, we further provide an algorithm, in which agents work in parallel, that achieves these bounds with limited communication.

Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021), PMLR 161:247–256.

We consider a setup where M reinforcement learning agents interact with M identical environments or Markov Decision Processes (MDPs) [Sutton and Barto, 2018, Puterman, 1994]. Similar to the examples considered above, the M environments are independent. We assume that there also exists a central coordinator (server), although we suggest a method to relax this assumption. All the agents report their experiences to the central coordinator which then computes and shares a policy with the agents. We consider that the central coordinator uses a model based algorithm to compute the policy and each agent stores the number of visitations made to any particular (state, action, next state) tuple. The central coordinator also shares the total visitations to each state-action pair back to all the agents.

Based on this setup, we provide a novel communication efficient DIST-UCRL algorithm. In DIST-UCRL algorithm, the agents communicate with the central coordinator whenever any agent visits any state action pair s, a in the current epoch (since last synchronization) a fraction $(1/M)$ of the total visitation counts in s, a till last synchronization, instead of every $N \geq 1$ time step. Using this synchronization strategy, for an MDP with diameter D, number of states S, and number of actions A, using DIST-UCRL algorithm, we bound the cumulative regret of M agents for T time horizon scales as $O(DS\sqrt{MAT})$ using only $O(MAS\log_2(MT))$ synchronization steps[1]. Note that DIST-UCRL not only achieves the lower bound regret scaling ($\tilde{O}(\sqrt{MT})$) but also with limited communication.

To obtain our results, we also derive a concentration bound on M independent Martingale sequences which can be of independent interest. To the best of our knowledge, this is the first work in the direction of obtaining the performance guarantees using regret analysis for a setup where M agents interact and collaborate.

We also evaluate our algorithms empirically. We run the proposed DIST-UCRL algorithm in multiple environments. We compare the DIST-UCRL algorithm against the modified UCRL algorithm, following which the agents communicates after every time step. We show that DIST-UCRL algorithm obtain similar regret bound with reduced communication.

The rest of the paper is organized as follows. Section 2 summarizes the key related works. Section 3 describes the complete system model. Section 4 describes the DIST-UCRL algorithm, and the regret guarantees of the algorithm are provided in Section 5. Section 6 describes a modification of UCRL2, MOD-UCRL2, and its regret guarantees. Section 7 tests the proposed algorithms in multiple environments, and Section 8 concludes the paper with some possible directions for future works.

[1]We use the term communication round and synchronization step interchangeably.

2 RELATED WORKS

Optimal planning using Markov Decision Processes has seen numerous significant contributions in history. Many algorithms have been proposed to find the optimal policies. The major algorithms include Q-learning, and policy iteration to find optimal policies [Howard, 1960, Puterman, 1994, Bertsekas, 1995, Sutton and Barto, 2018]. Fundamentally, the algorithms work by calculating the utility of taking certain actions in states that maximizes the utility of the state. These algorithms provide optimal policy when the transition probabilities of the Markov Decision Process is known or an ϵ-optimal policy if using iterative algorithms [Puterman, 1994].

A large body of work is also done in the setup where the transition probabilities are not known. The model-based algorithms work by reducing the number samples required to obtain a close estimate of transition probabilities [Bartlett and Tewari, 2009, Jaksch et al., 2010]. The model-free algorithms work by directly estimating the utilities obtained by taking an action in a state [Jin et al., 2018]. These algorithms apply optimism in the face of uncertainty principle to find a model near the empirical estimates that provides the highest reward. There are also algorithms that sample the transition probabilities using posterior sampling and obtain regret bound [Ian et al., 2013, Agrawal and Jia, 2017]. The algorithms suggest that using parallel agents interacting independent and identical environments will provide tighter concentration inequalities and hence will help in reducing the regret.

In the domain of multiple agents using RL, most of the work is done where the agents interact with a dependent environment and the decision of one agent impacts all the other agents. This area is known as Multi-Agent Reinforcement Learning (MARL) [Gupta et al., 2017, Zhang et al., 2018]. The agents may cooperate with each other, for example, in the case of autonomous vehicles yielding on a busy road. Or the agents may also compete with each other, for example, in the case of car racing where only one car may win. Although, we do have multiple agents in our setup, the environments are independent. Hence, we will refrain from using MARL terminology in this paper.

With the introduction of deep learning to the field of reinforcement learning, many "Deep RL" algorithms have been provided to minimize the sample complexity to find the optimal policies [Mnih et al., 2013, Schulman et al., 2015, Mnih et al., 2016, Schulman et al., 2017, Haarnoja et al., 2018]. Recently, various algorithms are using parallel actors to learn better policies using deep reinforcement learning [Nair et al., 2015, Clemente et al., 2017, Horgan et al., 2018, Espeholt et al., 2018, Assran et al., 2019]. These algorithms consist of parallel agents that share the entire sequence of (state, action, reward, next state) tuples after every n epochs, and update a common neural network with the gra-

dients computed from possibly parallel learners. Compared to these works, we consider a model based setup to obtain regret guarantees with only $O(MAS\log_2(MT))$ number of synchronization rounds with a common controller learning the policy.

Recently, in the area of Bandits [Lattimore and Szepesvári, 2020], there has been a thrust on distributed bandits with reduced number of communication rounds. Various algorithms have been proposed to minimize the regret for setups where the agents synchronize a central coordinator [Kanade et al., 2012, Hillel et al., 2013, Wang et al., 2019, Dubey and Pentland, 2020]. Recently, Wang et al. [2019] showed that it is possible to obtain optimal regret guarantees with number of rounds that are independent of T in stochastic Multi-Armed Bandits and Linear Bandits. Further, Dubey and Pentland [2020] considered a linear bandit setup and aimed to protect the privacy of the agents collaborating along with minimizing the number of communications rounds. Moreover, [Chawla et al., 2020, Sankararaman et al., 2019] considered a gossiping setup to communicate only the index of the best arm, thus reducing not only the number of communication rounds, but also the number of bits in each communication round. Similar to the bandit setup, we also attempt to find rigorous regret guarantees of the DIST-UCRL algorithm and a bound on the number of communication rounds.

3 SYSTEM MODEL

Let $[K]$ be the set of K elements, or $[K] = \{1, 2, \cdots, K\}$. We consider an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \bar{r})$, where $\mathcal{S} = [S]$ is the set of finite states and $\mathcal{A} = [A]$ is the set of finite actions. P denotes the transition probabilities, *i.e.*, on taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, the next state $s' \in \mathcal{S}$ follows distribution $P(\cdot|s, a)$. Also, on taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, an agent receives a stochastic reward r drawn from a distribution over $[0, 1]$ with mean $\bar{r}(s, a)$.

We consider M agents in the system, each interacting with M identical environments. Let $i \in [M]$ be the indexing for agents and their corresponding environment. For an agent i, at time step t, let $s_{i,t}$ denote the state of the agent, $a_{i,t}$ be the action taken by the agent, and $r_{i,t}$ denote the reward obtained by the agent on taking action $a_{i,t}$ in state $s_{i,t}$. We assume that the M environments are independent. Mathematically, $\forall\, i \in [M]$ and $\forall\, t \geq 1$, we have,

$$\mathbb{P}(s_{i,t+1}|s_{1,t}, a_{1,t}, \cdots, s_{M,t}, a_{M,t},) = \mathbb{P}(s_{i,t+1}|s_{i,t}, a_{i,t}),$$

and we take a similar assumption over the rewards $r_{i,t}$ as well. This means, the distribution of the next state $s_{i,t+1}$ and the rewards $r_{i,t}$ of agent $i \in [M]$ are conditioned only on the knowledge of the current state and action $(s_{i,t}, a_{i,t})$ of the agent i and is independent of the other states and actions of the remaining agents.

Let policy $\pi : \mathcal{S} \to \mathcal{A}$ be a function to determine which action to select in state $s \in \mathcal{S}$. Note that each policy induces a Markov Chain on the states $\mathcal{S}$ with transition probabilities $P_{s,s} = P(\cdot|s, \pi(s))$. After defining a policy, we can now define the diameter of the MDP $\mathcal{M}$ as:

Definition 1 (Diameter)**.** *Consider the Markov Chain induced by the policy π on the MDP $\mathcal{M}$. Let $T(s'|\mathcal{M}, \pi, s)$ be a random variable that denotes the first time step when this Markov Chain enters state s' starting from state s. Then, the diameter of the MDP $\mathcal{M}$ is defined as:*

$$D(\mathcal{M}) = \max_{s' \neq s} \min_{\pi} \mathbb{E}\left[T(s'|\mathcal{M}, \pi, s)\right] \tag{1}$$

We assume that the MDP $\mathcal{M}$ has a finite diameter which means that there is a policy, such that following that policy all $s \in \mathcal{S}$ communicate with each other.

Any agent $i \in [M]$ starting from an initial random state $s_{i,1} = s$ follows an algorithm $\mathscr{A}$ till T time steps to collect a cumulative reward of R_i. Also, let ρ_i denote the average reward of the agent following algorithm $\mathscr{A}$. Or,

$$R_i(\mathcal{M}, \mathscr{A}, s, T) = \sum\nolimits_{t=0}^{T} r_{i,t} \tag{2}$$

$$\rho_i(\mathcal{M}, \mathscr{A}, s) = \lim_{T\to\infty} \frac{1}{T} \sum\nolimits_{t=0}^{T} \mathbb{E}\left[r_{i,t}\right] \tag{3}$$

Let there be an algorithm $\mathscr{A}$ which always selects action according to a stationary policy $\pi : \mathcal{S} \to \mathcal{A}$. Then, we denote $\rho(\mathcal{M}, \mathscr{A}, s) = \rho(\mathcal{M}, \pi, s)$. The optimal average reward does not depend on the state [Puterman, 1994, Section 8.3.3], and hence for the optimal policy π^* which maximizes the average reward $\rho(\mathcal{M}, \pi, s)$ we have,

$$\rho^*(\mathcal{M}) \coloneqq \rho^*(\mathcal{M}, s) \coloneqq \max\nolimits_{\pi} \rho(\mathcal{M}, \pi, s). \tag{4}$$

Further, the optimal average reward satisfies [Puterman, 1994, Theorem 8.4.7],

$$\rho^* + v(s) = \bar{r}(s, a^*) + \sum\nolimits_{s'} P(s'|s, a^*)v(s')\ \forall s \in \mathcal{S} \tag{5}$$

where $a^* = \pi^*(s)$, and $v(s)$ is called the bias of the state s and it denotes the extra reward obtained from starting in the state s. Note that $v(s)$ is not unique since if $v(s)\ \forall\ s$ satisfy Equation (5), then so does $v(s) + c$ for all $c \in \mathbb{R}$ and hence, the bias is translation invariant.

We aim to maximize the cumulative reward collected by all the agents. Hence, we want to develop an algorithm that, starting from no knowledge about the system, learns a policy that minimizes the regret. The regret of an algorithm $\mathscr{A}$, for starting state s, and running for time T, is defined as:

$$\Delta(\mathcal{M}, \mathscr{A}, s, T) := \rho^* MT - \sum\nolimits_{i\in[M]} R_i(\mathcal{M}, \mathscr{A}, s, T)$$

We will now present our algorithm DIST-UCRL that uses upper confidence bounds to bound the regret $\Delta(\mathcal{M}, \text{DIST-UCRL}, s, T)$ with high probability with only $O(MAS\log_2(MT))$ communication rounds.

Algorithm 1 DIST-UCRL at agent i

Input: S, A, M
Set parameters $P_i(s,a,s') = 0 \forall (s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ and $\hat{r}_i(s,a) = 0 \forall (s,a) \in \mathcal{S} \times \mathcal{A}$.
for Epochs: $k = 1, 2, \cdots$ **do**
  Set $\nu_{i,k}(s,a) = 0 \forall (s,a) \in \mathcal{S} \times \mathcal{A}$.
  $\pi_k, N_k = \text{SYNCHRONIZE}(P_i, \hat{r}_i, t)$
  while $\nu_{i,k}(s,a) < \max(1, N_k(s,a))/M \forall (s,a)$ and Synchronization not requested **do**
    Play action $a_{i,t} = \pi_k(s_{i,t})$, observe reward r_t and next state $s_{i,t+1}$.
    Set $\nu_{i,k}(s_{i,t}, a_{i,t}) = \nu_{i,k}(s_{i,t}, a_{i,t}) + 1$, $P_i(s_{i,i,t}, a_{i,t}, s_{t+1}) = P_i(s_{i,t}, a_{i,t}, s_{i,t+1}) + 1$, $\hat{r}_i(s_{i,t}, a_{i,t}) = \hat{r}_i(s_{i,t}, a_{i,t}) + r_t$.
    Set $t = t + 1$.
  end while
  Request synchronization
end for

Algorithm 2 SYNCHRONIZE at central node

Input: $P_i, \hat{r}_i$ from all agents $i \in [M]$, t.
for $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**
  Set $N(s,a) = \sum_i \sum_{s'} P_i(s,a,s')$.
  Set $\hat{p}(s,a,s') = \frac{\sum_i P_i(s,a,s')}{\max\{1, N(s,a)\}}$
  Set $\hat{\bar{r}}(s,a) = \frac{\sum_i \hat{r}_i(s,a)}{\max\{1, N(s,a)\}}$
  Set $\tilde{r}(s,a) = \hat{\bar{r}}(s,a) + \sqrt{\frac{7 \log(2MSAt)}{2 \max\{1, N(s,a)\}}}$
  Set $d(s,a) = \sqrt{\frac{14S \log(2MAt)}{\max\{1, N(s,a)\}}}$
end for
Set π = EXTENDED VALUE ITERATION$(\hat{p}, d, \tilde{r}, \frac{1}{\sqrt{Mt}})$
Return π, N

4 DIST-UCRL ALGORITHM

We consider that each agent runs an instance of the DIST-UCRL algorithm. The DIST-UCRL algorithm running at agent i is described in Algorithm 1. The algorithm proceeds in epochs indexed as $k = 1, 2, \cdots$. The start of every epoch also coincides with the synchronization step where every agent communicates with the central node to share data and update policies. This also implies that the number of synchronization rounds required by the algorithm are the same as the number of epochs for the algorithm runs.

Algorithm 1 running at agent i, maintains two counters, $\nu_{i,k}(s,a)$ and $P_i(s,a,s')$. $\nu_{i,k}(s,a)$ counts the number of visitations to state action pair (s,a) in epoch k, and $P_i(s,a,s')$ counts the instances when the agent moves to state s' on taking action a in state s. The agent also stores $\hat{r}_i(s,a)$ as the cumulative reward obtained in (s,a).

Let $N_k(s,a) = \sum_{i=1}^{M} \sum_{k'=1}^{k-1} \nu_{i,k}(s,a)$ be the total number of visitations till the start of epoch k for all agents. And hence, $N_k(s,a) = 0$ at $k = 1$ for all $s, a \in \mathcal{S} \times \mathcal{A}$. At the start of every epoch $k \geq 1$, the agents obtains the policy for epoch k and total visitations to any state action pair $N_k(s,a)$. We will denote the time step at which epoch k start and agents synchronize of the k^{th} time with t_k. At $t = 1$, the algorithm synchronizes all the agents for the very first time, or $t_1 = 1$. Later, a new epoch is triggered whenever any of the agent requests for synchronization[2]. An agent i requests synchronization whenever $\nu_{i,k}(s,a)$ becomes at least $1/M$ of $N_k(s,a)$ for any state action pair. We assume that every agent is able to receive the synchronization signal instantly and stop further processing of the current epoch. The algorithm calls SYNCHRONIZE algorithm every time after a new epoch starts and updates the policy π_k and $N_k(s,a)$ values. Every agent now selects actions according to the policy π_k in the epoch k.

The SYNCHRONIZE algorithm is described in Algorithm 2. This algorithm calculates the estimates of the transition probability $\hat{p}(\cdot|s,a)$ and the mean rewards $\hat{\bar{r}}(s,a)$ using the samples from all the M agents. We now consider a set of all plausible MDPs $\mathscr{M}(t)$ that exist in the neighborhood of the estimated MDP $\widehat{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \hat{p}, \hat{\bar{r}})$. The mean rewards $r'(s,a)$ and the transition probabilities $p'(s,a)$ for all the MDPs in the set $\mathscr{M}(t)$ satisfies:

$$|\hat{\bar{r}}(s,a) - r'(s,a)| \leq \sqrt{\frac{7 \log(2MSAt)}{2 \max\{1, N(s,a)\}}} \quad (6)$$

$$\|\hat{p}(\cdot|s,a) - p'(\cdot|s,a)\|_1 \leq \sqrt{\frac{14S \log(2MAt)}{\max\{1, N(s,a)\}}} \quad (7)$$

After obtaining $\mathscr{M}(t)$, Algorithm 2 calls the EXTENDED VALUE ITERATION algorithm which then computes the optimal policy for the optimistic MDP $\tilde{\mathcal{M}}_t$ in the set $\mathscr{M}(t)$. The optimistic MDP satisfies $\rho^*(\tilde{\mathcal{M}}) = \sup_{\mathrm{M} \in \mathscr{M}(t)} \rho^*(\mathrm{M})$. As described in [Jaksch et al., 2010], it is not trivial to directly find the optimistic MDP in $\mathscr{M}(t)$. Hence, we consider an extended MDP $\mathcal{M}_t^+$ which is constructed with the same state space and a continuous action space $(a, q(\cdot|\cdot,a)) \in \mathcal{A} \times \mathscr{P}_t$, where $\mathscr{P}_t$ is the set of transition probabilities for action $a \in \mathcal{A}$ that satisfies Equation (7). When $\mathscr{M}(t)$ contains the true MDP $\mathcal{M}$, the diameter of the extended MDP $\mathcal{M}_t^+$ is bounded by D as the policy for which all states communicate with each other for MDP $\mathcal{M}$ also ensures that all states communicate in the extended MDP $\mathcal{M}_t^+$ as well.

The EXTENDED VALUE ITERATION algorithm (Algorithm 3) follows the design of the Extended Value Iteration of UCRL2 algorithm by Jaksch et al. [2010]. As described by Jaksch et al. [2010], EXTENDED VALUE ITERATION (EVI) obtains a policy π that is ϵ-optimal for the extended MDP $\mathcal{M}_t^+$ and in turn the optimistic MDP $\tilde{\mathcal{M}}$. Algorithm 3 calculates the values of the states of the extended MDP

[2]The synchronization request can be sent to server, which will in turn pause and synchronize the entire system.

Algorithm 3 EXTENDED VALUE ITERATION

Input: $\hat{p}, d, \tilde{r}, \epsilon$.
Set $u_0(s,a) = 0, u_1(s) = \max_a \tilde{r}(s,a), i = 1$.
Set $\pi(s) = \arg\max_a \tilde{r}(s,a)$
while $\max_s\{u_i(s) - u_{i-1}(s)\} - \min_s\{u_i(s) - u_{i-1}(s)\} \geq \epsilon$ **do**
  Sort $s'_1, \cdots, s'_S$ such that $u_i(s'_1) \geq \cdots \geq u_i(s'_S)$.
  Set $p(s_1) = \min\{1, \hat{p}(s'|s_1,a) + d(s_1,a)/2\}$
  Set $p(s_n) = \hat{p}(s'|s_n,a), n = 2,3,\cdots,S$.
  Set $l = 1$
  while $\sum_s p(s) > 1$ **do**
    Set $p(s_l) = \max\{0, 1 - \sum_{s_n \neq s_l} p(s_n)\}$
    Set $l = l - 1$
  end while
  $i = i + 1$
  $u_i(s) = \max_a \{\tilde{r}(s,a) + \sum_{s'} p(s') u_{i-1}(s')\}$
  $\pi(s) = \arg\max_a \{\tilde{r}(s,a) + \sum_{s'} p(s') u_{i-1}(s')\}$
end while
Return π

$\mathcal{M}_t^+$. The extended value iteration calculates the utilities of the states and the actions that achieve this utility. Note that unlike the extended value iteration in UCRL2 algorithm, we consider Algorithm 3 to be converged when we have $\max_s(u_{i+1} - u_i(s)) - \min_s(u_{i+1} - u_i(s)) \leq \epsilon = 1/\sqrt{Mt}$. This is because we now have M times more samples till any time step t as compared to the UCRL algorithm. The EVI, at start of epoch k, returns a policy π_k that satisfies $\rho(\tilde{\mathcal{M}}, \pi_k) \geq \rho^*(\tilde{\mathcal{M}}_{t_k}) - 1/\sqrt{Mt_k}$.

We also note that the central controller is not necessarily required if the agents are in a completely connected network, they can share there data with each other and run the algorithms for the central controller by themselves. Further, the completely connected assumption can also be relaxed by considering a setup where all agents forward the messages they get. This allows the broadcast of the information. Hence, the proposed algorithm can be generalized to any network structure as long as all the agents are connected via some path.

5 RESULTS FOR DIST-UCRL

After describing the algorithm, we now bound the regret of the DIST-UCRL algorithm. We show that the regret bound holds with high probability. We bound the regret incurred by the DIST-UCRL algorithm in the form of following theorem.

Theorem 1. *For a MDP $\mathcal{M} = ([S],[A],P,r)$ with diameter D, for any starting state s, the regret of the* DIST-UCRL *algorithm, running on M agents for T time steps, is upper bounded with probability at least $1 - \frac{1}{(MT)^{5/4}}$ as:*

$$\Delta(\mathcal{M}, \text{DIST-UCRL}, s, T) \leq \tilde{O}(DS\sqrt{MAT}) \quad (8)$$

where $\tilde{O}$ hides the poly-log terms in M, S, A, and T.

We let m be the total number synchronizations done by agents running DIST-UCRL algorithm till time T. Then, we bound m deterministically in the following theorem.

Theorem 2. *The total number of communication rounds m for dist-UCRL2 up to step $T \geq SA/M$ is upper bounded as*

$$m \leq 1 + 2MAS + MAS\log_2(MT) \quad (9)$$

Proof. We use the fact that when $\nu_{i,k}(s,a) \geq N_k(s,a)/M$ for some state action pair (s,a) and for some agent i, the total visitation count $N_{k+1}(s,a)$ is at least $N_k(s,a) + \nu_{i,k}(s,a) = N_k(s,a)(1 + \frac{1}{M})$. This gives an exponential growth for the total visitation count of any state action pair. Also, since the total visitation count for all state action pairs is upper bounded by MT, using Jensen's inequality we bound the number of epochs in logarithmic order of MT. A complete proof is provided in the supplementary material. □

We now state the lemmas required for the proof of the Theorem 1. The first three lemmas are used to handle the stochastic nature of the algorithm and environment. The first lemma provides concentration bounds on the ℓ_1-deviation of the transition probabilities $\hat{p}(\cdot|s,a)$ for any (s,a).

Lemma 1. *The ℓ_1-deviation of the true distribution and the empirical distribution using n samples, over the next states given the current state s and action a is bounded by*

$$\mathbb{P}\left(\|\hat{p}(\cdot|s,a) - P(\cdot|s,a)\|_1 \geq \epsilon\right) \leq 2^S \exp\left(-\frac{n\epsilon^2}{2}\right) \quad (10)$$

Proof. The proof follows on the lines of [Weissman et al., 2003, Theorem 2.1], using the distribution as transition probabilities. □

Lemma 2 (Hoeffding's Inequality, [Hoeffding, 1994])**.** *Let $\{X_t\}_{t=1}^T$ be i.i.d. random variables in [0,1]. Then, we have,*

$$P(\sum_{t=1}^{T} X_{i,t} \geq \epsilon) \leq \exp\left(-\frac{2\epsilon^2}{T}\right) \quad (11)$$

The next lemma provides concentration bounds on the sum of M independent Martingale sequences for length T.

Lemma 3. *Let $\{X_{i,t}\}_{t=1}^T$ be a zero-mean Martingale sequence for $i = 1,\cdots,M$ adapted to filtration $\{\mathcal{F}_t\}_{t=0}^T$. Then, if $\{X_{i,t}\}_{t=1}^T$ and $\{X_{j,t}\}_{t=1}^T$ are independent for all $i \neq j$ and $|X_{i,t}|X_{i,t-1}| \leq c$ for all i,t, we have,*

$$P\left(\sum_{t=1}^{T}\sum_{i=1}^{M} X_{i,t} \geq \epsilon\right) \leq \exp\left(-\frac{2\epsilon^2}{MTc^2}\right) \quad (12)$$

Proof Sketch. We prove this lemma similar to the proof of Azuma-Hoeffding's Inequality [Hoeffding, 1994]. A detailed proof is provided in the supplementary material. □

We now bound the growth rate of the total number of visitations to state action pair (s,a), $\sum_{i=1}^{M}\nu_{i,k}(s,a)$, in any epoch k. If the total visitations are large, then the agents will incur large regret from a possibly sub-optimal policy. Hence, we have the following lemma:

Lemma 4. *For any epoch k, we have,*

$$\sum\nolimits_{i=1}^{M}\nu_{i,k}(s,a)\leq N_k(s,a)+M-1 \tag{13}$$

Proof. Note that agent i requests for synchronization, and triggers a new epoch, whenever $\nu_{i,k}(s,a) = \lceil N_k(s,a)/M\rceil \leq N_k(s,a)/M+(M-1)/M$. Summing over all the agents i gives the bound. □

Lemma 5 (Lemma 19 [Jaksch et al., 2010])**.** *For any sequence of number $z_1\leq z_2\leq\cdots\leq z_n$ with $z_k\leq\sum_{k'=1}^{k-1}z_{k'}=:Z_k$, we have,*

$$\sum_{k=1}^{n}\frac{z_k}{Z_k}\leq(\sqrt{2}+1)Z_n \tag{14}$$

The last lemma states that the span of the bias of the optimal policy defined as $\max_s v(s)-\min_s v(s)$ is bounded by the diameter D.

Lemma 6 (Remark 8 from Jaksch et al. [2010])**.** *The span of the bias $v:\mathcal{S}\to\mathbb{R}$ of the optimal policy π for any MDP $\mathcal{M}$ is upper bounded by its diameter $D(\mathcal{M})$, or,*

$$sp(v)=\max_s v(s)-\min_s v(s)\leq D(\mathcal{M}) \tag{15}$$

After stating all the necessary lemmas, we are now ready to prove the regret bound of the DIST-UCRL algorithm or $\Delta(\mathcal{M},\text{DIST-UCRL},s,T)$. We provide a detailed sketch here and provide the complete proof in the supplementary material.

Proof Sketch of Theorem 1. We break the regret expression, into 4 different sources of regrets as:

1. Regret from deviating from expected reward: Note that regret compares the expected optimal gain ρ^* with the observed rewards $r_{i,t}$. Since, $r_{i,t}$ is a random variable between $[0,1]$, the agents suffer a regret if the observed rewards are lower as compared to the mean. Hence, we use Hoeffding's inequality [Hoeffding, 1994] to bound the regret generated by the randomness of the observed rewards. This gives us regret bounded by $\tilde{O}(\sqrt{MT})$.

2. Regret from deviating from the expected next state: The algorithm, when transitioning to the next state, expects a bias given the current state. However, the bias of the realized state may be different and even lower from the expected bias. Hence, we bound the deviation from the expected bias as the algorithm moves to states. Starting from state s, the deviation process is modelled as a zero-mean Martingale sequence of the states visited by an agent i. Also, we have M independent agents interacting with M independent environment. We use Lemma 3 to bound the total deviation of the realized bias and the expected bias. This gives us regret bounded by $\tilde{O}(D\sqrt{MT})$.

3. Regret from not optimizing for the true MDP: For epoch k, we run an $\frac{1}{\sqrt{Mt_k}}$-optimal policy for an optimistic MDP from the set of MDPs $\mathscr{M}(t_k)$ for $\sum_{i\in[M]}\sum_{s,a}\nu_{i,k}(s,a)$. The transition probabilities and the mean rewards of MDPs in $\mathscr{M}(t_k)$ satisfy Equation (7) and Equation (6). When the true MDP $\mathcal{M}$ lies in $\mathscr{M}(t_k)$, we bound the regret because of the incorrect MDPs by the product of the diameter of the set $\mathscr{P}_{t_k}$, the diameter D of the MDP $\mathcal{M}$, and the number of visitations to any state action pair s,a in epoch k using Lemma 6. All we need to do now is to bound the sum of the product for all epochs. We do so by using Lemma 4 and Lemma 5. This gives us regret bounded by $\tilde{O}(DS\sqrt{MAT})$.

4. Regret when the estimated MDP is far from the true MDP $\mathcal{M}$: We use Lemma 1 to bound the ℓ_1 distances of the estimated transition probability and the true transition probability given any state action pair. Further, we also use Hoeffding's inequality (Lemma 2) to bound the distance of the true mean rewards and the estimated rewards. Taking union bounds over all time steps till T, we obtain the bound for all possible values of $N(s,a)$. Further, taking union bounds over all state and actions provide the desired concentration bounds for all states and actions. Finally, we take the union bounds for all values of $t\geq(MT)^{1/4}$. This gives the total bound on probability that the true MDP $\mathcal{M}$ lies in $\mathscr{M}(t_k)$ as:

$$\mathbb{P}\left(\mathcal{M}\notin\mathscr{M}(t)\right)\leq\frac{1}{(MT)^{5/4}}\forall\, t\geq(MT)^{5/4}. \tag{16}$$

Now summing over all the regret sources, we get the regret bound of the DIST-UCRL algorithm. □

The proof of the Theorem 1 and Theorem 2 suggests that the analysis could potentially be extended to various other algorithms that follow the epoch completion condition from the UCRL-2 Algorithm of [Jaksch et al., 2010].

6 NAIVE APPROACH: MOD-UCRL2 FOR MULTIPLE AGENTS

The proposed DIST-UCRL algorithm does not require the agents to work sequentially, and hence the M agents can truly work in parallel. For comparison of the proposed algorithm and completeness, we also consider an extension

of UCRL2 algorithm by Jaksch et al. [2010] for M parallel agents which we call MOD-UCRL2. In the MOD-UCRL2 algorithm, we assume that all the agents communicate to a centralized server at each time step t, and the server decides the actions for the agents at each time t. Thus, the communication rounds for MOD-UCRL2 algorithm is $O(T)$, while the regret analysis is not apriori clear, and is the focus of this section. We also note that even though this algorithm is an easy extension of UCRL2, the analysis of regret is not straightforward, and uses the approach that was used to prove the regret guarantees of DIST-UCRL because of the presence of M agents.

At every time step t, every agent $i \in [M]$ observes its state $s_{i,t}$ and sends it to the server. The server after receiving all the states, process the requests sequentially in the order $s_{1,t}, s_{2,t}, \cdots, s_{M,t}$. The central server runs an instance of the UCRL2 algorithm with state sequence $s_{1,1}, s_{2,1}, \cdots, s_{M,1}, s_{1,2}, \cdots$, and the corresponding action sequence $a_{1,1}, a_{2,1}, \cdots, a_{M,1}, a_{1,2}, \cdots$. The UCRL2 algorithm, running at the server, proceeds in epoch with epoch 1 starting with $(i,t) = (1,1)$. We consider an epoch k contains observations for $\{(\underline{i}_k, \underline{t}_k), (\underline{i}_k, \underline{t}_k) + 1, \cdots, (\bar{i}_k, \bar{t}_k)\}$ for some $\underline{i}_k, \underline{t}_k, \bar{i}_k, \bar{t}_k$. The server maintains counters $\nu_k(s,a)$ denoting the number of times a state action pair s, a is visited in the epoch k, and counter $N_k(s,a)$ denoting the number of times a state action pair s, a is visited before the start of the epoch k as:

$$\nu_k(s,a) = \sum_{(i,t)=(\underline{i}_k,\underline{t}_k)}^{(\bar{i}_k,\bar{t}_k)} \mathbf{1}\{s_{i,t} = s, a_{i,t} = a\} \quad (17)$$

$$N_k(s,a) = \sum_{k'=1}^{k-1} \nu_{k'}(s,a) \quad (18)$$

The server updates an epoch k whenever $\nu_k(s,a) = \max\{1, N_k(s,a)\}$ for some s, a. Following the UCRL2 algorithm, the server updates the policy at the beginning of every epoch using the observations collected till the beginning of the epoch. The complete algorithm is provided in the supplementary material.

Note that in the MOD-UCRL2 algorithm, the agents only behaves as interfaces to the M independent environments, and is essentially not practical because of the sequential interface. In the following result, we formally state the result that the regret is upper bounded by $\tilde{O}(DS\sqrt{MAT})$.

Theorem 3. *For a MDP $\mathcal{M} = ([S], [A], P, r)$ with diameter D, for any starting state s, the regret of the* MOD-UCRL2 *algorithm, running on M agents for T time steps, is upper bounded with probability at least $1 - \frac{1}{(MT)^{5/4}}$ as:*

$$\Delta(\mathcal{M}, \text{MOD-UCRL2}, s, T) \leq \tilde{O}(DS\sqrt{MAT})$$

where $\tilde{O}$ hides the poly-log terms in M, S, A, and T.

Proof Outline. Similar to the proof of Theorem 1, we again consider the four sources of regret. Then, breaking the regret into episodes when the MOD-UCRL2 algorithm updates policy, we obtain the required bound. The complete proof is provided in the supplementary material. □

7 EVALUATIONS

In this section, we analyze the performance of the proposed DIST-UCRL algorithm empirically. We test the DIST-UCRL algorithm in multiple environments and also vary the number of agents in all the cases to study the regret growth with respect to M. Further, we also evaluate the average number of communication steps used by the DIST-UCRL algorithm till time step T.

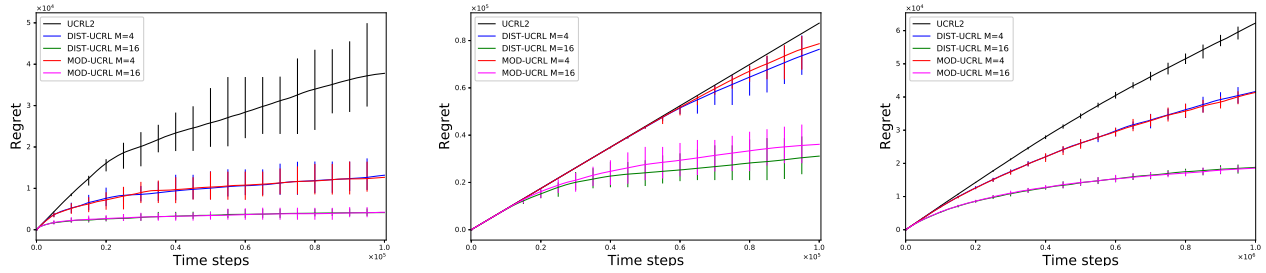
We first run the DIST-UCRL algorithm for the RiverSwim environment, which is a standard benchmark for model-based RL algorithm [Ian et al., 2013, Tossou et al., 2019] with 6 states and 2 actions. Next, we construct an extended RiverSwim environment with 12 states and 2 actions. Finally, we use a Grid-World environment [Sutton and Barto, 2018] for a 7×7 grid which amounts to 20 states and 4 actions.

We compare the DIST-UCRL algorithm against the MOD-UCRL2 algorithm. We also compare with the standard UCRL2 algorithm for $M = 1$. Note that for the case of $M = 1$, both, DIST-UCRL algorithm and the MOD-UCRL2 algorithm reduces to the UCRL2 algorithm.

We run 50 independent iterations of the algorithm. We plot the average per-agent regret, $\Delta(\mathcal{M}, \text{DIST-UCRL}, s, T)/M$, over the 50 iterations and the error bars for both DIST-UCRL and MOD-UCRL2 algorithm in Figure 1. We vary the number of agents M as $1, 4$, and 16 to reduce clutter in figures. From Figure 1(a) and Figure 1(c), we note that the per-agent regret decreases approximately by a factor of 2 for every 4-fold increase in the number of agents. This is expected and was indeed the goal.

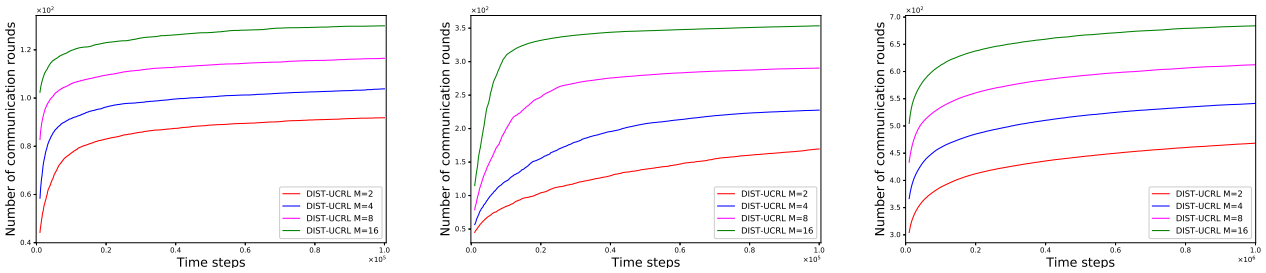
Note that in Figure 1(b), the per-agent regret falls from being linear in UCRL2 ($M = 1$) to significantly sublinear as the number of agents were increased to $M = 4$ and $M = 16$. For the extended RiverSwim environment with 12 states, the diameter is approximately 5×10^4. Hence, the available time horizon of $T = 10^5$ was not sufficient for UCRL2. However, distributed RL algorithms still observed a sub-linear regret by collating their knowledge. This further demonstrates the advantage of deploying multiple parallel agents in complex environments to enhance exploration.

We also empirically evaluate the number of communication rounds required by the DIST-UCRL algorithm. Note that the number of communications for MOD-UCRL2 algorithm is same as the total length for which the algorithm runs. Hence, we only plot the number of communications rounds required for the DIST-UCRL algorithm. We vary the number of agents M in multiples for 2 and starting from $M = 2$



(a) RiverSwim environment (b) RiverSwim-12 environment (c) Gridworld environment

Figure 1: Average cumulative regret per agent under various communication strategies. The empirical regret of the DIST-UCRL algorithm and MOD-UCRL2 algorithm are almost identical which is expected from the regret analysis of the two algorithms.



(a) RiverSwim environment (b) RiverSwim-12 environment (c) Gridworld environment

Figure 2: Total number of communication rounds required for the DIST-UCRL algorithm for multiple number agents across various environments.

agents. We again run the experiment for 50 independent iterations and plot the average number of synchronization rounds till time step t.

We plot the number of communication rounds of the DIST-UCRL algorithm in Figure 2. We observe that the number of the synchronization rounds increase very slowly with t. Further, as suggested from Theorem 2, the number of synchronization rounds are increasing in M. However, we note that for large values of t, the increase in the number of communication rounds is sub-linear. This is because for the bound on the number of communication rounds, we take pessimistic estimates on the increase on the visitation counts $N_k(s,a)$ where only one agent i updates total $N_k(s,a)$ by a factor of $(1+1/M)$ only when the agent i triggers the synchronization round for s,a. However, in practice, $N_k(s,a)$ grows faster than $(1+1/M)$, as the synchronization rounds triggered for other state action pairs and other agents also contribute towards $N_k(s,a)$.

8 CONCLUSION

In this work, we considered the problem of simultaneously reducing the cumulative regret and the number of communication rounds between M agents. The M agents interact with M independent and identical Markov Decision Processes and share their data to learn the optimal policy faster. To this end, we proposed DIST-UCRL, an epoch-based algorithm, following which the agents communicate in the beginning of every epoch. The data collected from the M agents allows obtaining tighter deviation bounds and hence a smaller confidence set. Further, the agents trigger an epoch only after collecting sufficient samples in every epoch. This allows us to bound the regret of the DIST-UCRL algorithm as $\tilde{O}(DS\sqrt{MAT})$ and the number of communication rounds as $O(MAS\log_2(MT))$. To analyze our algorithm, we also provide a concentration inequality for M independent Martingale sequence of equal length which may be of an independent interest. We also evaluate the algorithm empirically and found that the average per-agent regret decreases as $O(1/\sqrt{M})$.

For comparison, we consider an extension of the UCRL2 algorithm for M parallel agents working in round-robin sequence, and denote this algorithm as MOD-UCRL2. We show that this algorithm also achieves the same regret bound as DIST-UCRL algorithm, while with $O(T)$ communication rounds. The regret guarantee required the techniques developed for DIST-UCRL. The evaluation results demonstrate the effectiveness of the proposed DIST-UCRL algorithm in achieving the same regret bound as MOD-UCRL2, while with lower communication. Thus, the proposed DIST-UCRL

algorithm can be utilized in training multiple parallel power starved devices using reinforcement learning.

References

Shipra Agrawal and Randy Jia. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1184–1194, 2017.

Abubakr O Al-Abbasi, Arnob Ghosh, and Vaneet Aggarwal. Deeppool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 20 (12):4714–4727, 2019.

Mahmoud Assran, Joshua Romoff, Nicolas Ballas, Joelle Pineau, and Michael Rabbat. Gossip-based actor-learner architectures for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Peter L Bartlett and Ambuj Tewari. Regal: a regularization based algorithm for reinforcement learning in weakly communicating mdps. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 35–42, 2009.

Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. 1995.

Ronshee Chawla, Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. The gossiping insert-eliminate algorithm for multi-agent bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 3471–3481. PMLR, 2020.

Alfredo V Clemente, Humberto N Castejón, and Arjun Chandra. Efficient parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1705.04862*, 2017.

Abhimanyu Dubey and AlexSandy' Pentland. Differentially-private federated linear bandits. *Advances in Neural Information Processing Systems*, 33, 2020.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pages 1407–1416. PMLR, 2018.

Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 66–83. Springer, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

Eshcar Hillel, Zohar Karnin, Tomer Koren, Ronny Lempel, and Oren Somekh. Distributed exploration in multi-armed bandits. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 1*, pages 854–862, 2013.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado van Hasselt, and David Silver. Distributed prioritized experience replay. In *International Conference on Learning Representations*, 2018.

Ronald A Howard. Dynamic programming and markov processes. 1960.

Junyan Hu, Hanlin Niu, Joaquin Carrasco, Barry Lennox, and Farshad Arvin. Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 2020.

Osband Ian, Van Roy Benjamin, and Russo Daniel. (more) efficient reinforcement learning via posterior sampling. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pages 3003–3011, 2013.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(4), 2010.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 4868–4878, 2018.

Varun Kanade, Zhenming Liu, and Božidar Radunović. Distributed non-stochastic experts. In *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 1*, pages 260–268, 2012.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *arXiv preprint arXiv:2002.00444*, 2020.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Kaushik Manchella, Abhishek K Umrawal, and Vaneet Aggarwal. Flexpool: A distributed model-free deep reinforcement learning algorithm for joint passengers and goods transportation. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, et al. Massively parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1507.04296*, 2015.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994. ISBN 0471619779.

Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. Social learning in multi agent multi armed bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–35, 2019.

Guillaume Sartoretti, Yue Wu, William Paivine, TK Satish Kumar, Sven Koenig, and Howie Choset. Distributed reinforcement learning for multi-robot decentralized collective construction. In *Distributed autonomous robotic systems*, pages 35–49. Springer, 2019.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Aristide C. Y. Tossou, Debabrota Basu, and Christos Dimitrakakis. Near-optimal optimistic reinforcement learning using empirical bernstein inequalities. *CoRR*, abs/1905.12425, 2019.

Yuanhao Wang, Jiachen Hu, Xiaoyu Chen, and Liwei Wang. Distributed bandit learning: Near-optimal regret with efficient communication. In *International Conference on Learning Representations*, 2019.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the l1 deviation of the empirical distribution. 2003.

Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. Fully decentralized multi-agent reinforcement learning with networked agents. In *International Conference on Machine Learning*, pages 5872–5881. PMLR, 2018.