Online Reinforcement Learning in Markov Decision Process Using Linear Programming

Vincent Leon and S. Rasoul Etesami

Abstract— **We consider online reinforcement learning in episodic Markov decision process (MDP) with unknown transition function and stochastic rewards drawn from some fixed but unknown distribution. The learner aims to learn the optimal policy and minimize their regret over a finite time horizon through interacting with the environment. We devise a simple and efficient model-based algorithm that achieves $\widetilde{O}(LX\sqrt{TA})$ regret with high probability, where L is the episode length, T is the number of episodes, and X and A are the cardinalities of the state space and the action space, respectively. The proposed algorithm, which is based on the concept of "optimism in the face of uncertainty", maintains confidence sets of transition and reward functions and uses occupancy measures to connect the online MDP with linear programming. It achieves a tighter regret bound compared to the existing works that use a similar confidence set framework and improves computational effort compared to those that use a different framework but with a slightly tighter regret bound.**

–*Keywords:* **Markov decision process, online learning, reinforcement learning, regret minimization, occupation measure, linear programming.**

I. INTRODUCTION

Sequential decision-making in an unknown environment is one of the central problems in many disciplines, such as control and game theory, operations research, and computer science [2]–[5]. In that regard, Markov decision process (MDP) provides a powerful paradigm for modeling and analyzing sequential decision-making in the face of uncertainty. In a standard discrete-time MDP, the system is in a certain state at each time, and the decision-maker has to choose an action based on the current state. Depending on the action taken, the decision-maker receives a reward from the environment, and the system transitions to the next state stochastically based on some state transition function. In reality, the decision-maker often may not have information about the underlying dynamics of the system and must learn them in real-time by executing different policies while interacting with the environment. Such problems are referred to as *online reinforcement learning*, and the decision-maker is called the learner or the player. MDPs and reinforcement learning have emerged in many applications such as robotic control [6], [7], game playing [8], partially observable opti-mal feedback control [9], [10], cyber-physical security [11], [12], healthcare [13], and among many others [5], [14].

In this paper, we study online reinforcement learning in episodic MDPs, where the learner's interaction with the environment proceeds in repeated episodes. In other words, the system "restarts" or gets to some initial state after a time period (episode), and learning takes place between episodes. This is a realistic model for many real scenarios. For example, each round of playing chess with an opponent can be viewed as an episode, and each episode contains multiple time steps for moving the chess pieces. Therefore, a chess player learns winning strategies after each round of the game and becomes more expert by means of rounds of play and learning.

Specifically, we consider an episodic MDP with finite X states, A actions, T episodes, and L steps in each episode. We devise a learning algorithm based on the optimism principle, which comprises two parts: model estimation and policy optimization. The first part estimates the model of MDP, and the second part optimizes the policy for implementation based on the estimated model. Our benchmark to measure the algorithm performance is *regret*, which is defined as the difference between the maximum expected cumulative reward that one can obtain given the MDP model and the expected cumulative reward returned by the algorithm (see Section II for a formal definition). The objective is to design a tractable reinforcement learning algorithm that achieves a regret sublinear in T and polynomial in other parameters.

A. Related Works

The concept of "optimism in the face of uncertainty" is a generally well-understood and strategic principle for decision-making over a fixed time horizon. It is widely applied in multi-armed bandit and reinforcement learning problems. Under such a principle, the algorithm maintains a plausible estimate of the model and chooses the action that yields the most optimistic reward. Earlier works such as UCRL algorithm of [15], UCRL2 algorithm of [16], and REGAL algorithm of [17] apply this principle and study a similar MDP model with unknown environment dynamics where the reward is randomly drawn from some unknown fixed distribution. These algorithms are composed of two parts: model estimation and policy optimization. UCRL2 algorithm achieves a regret bound of $\widetilde{O}(L^{\frac{3}{2}}X\sqrt{TA})$. UCBVI algorithm for episodic MDP, which is introduced in [18], improves this bound to $\widetilde{O}(L\sqrt{TXA})$ and achieves the best-known regret bound so far. UCBVI algorithm adopts a different optimism approach for reinforcement learning,

combining the upper confidence bound (UCB) approach with a Q-learning-type algorithm. In a more recent work [19], the authors propose a variant of model-free Q-learning algorithm that directly parameterizes and updates the value functions without explicitly modeling the environment. The Q-learning algorithm is combined with the UCB algorithm and achieves a regret of $\widetilde{O}(L^2\sqrt{TXA})$. Another notable algorithm-design principle is posterior sampling, a.k.a. Thompson Sampling. In [20], the authors present a posterior-sampling-based learning algorithm that achieves a regret bound of $\widetilde{O}(L\sqrt{TXA})$.

Recently, several works have extended the approach of confidence sets to adversarial MDPs. In adversarial MDP, the reward functions can change arbitrarily between episodes as if they were chosen by an adversary, whereas the environment dynamics still remain fixed. [21]–[23] tackle the reinforcement learning in episodic adversarial MDPs and apply online mirror descent on the space of occupancy measures instead of optimizing policies directly. Inspired by UC-O-REPS and UOB-REPS algorithms in [22] and [23] respectively, we adopt the concept of occupancy measure in our paper for non-adversarial setting.

B. Contributions

In this work, we devise a reinforcement learning algorithm for episodic MDPs which achieves a regret bound of $\widetilde{O}(LX\sqrt{TA})$ and requires at most $XA\log T$ times of model update and optimization by combining UCRL2 algorithm and UOB-REPS algorithm and extending their analysis. Moreover, the optimization problem for policy update is essentially a linear program and can be solved efficiently using standard linear programming solvers. Our contribution is twofold:

- Compared to the best regret bound of existing algorithms that adopt the same method based on confidence set, we reduce the regret bound from $\widetilde{O}(L^{\frac{3}{2}}X\sqrt{TA})$[1] in [16] to $\widetilde{O}(LX\sqrt{TA})$.
- Compared to the best regret bound of all existing algorithms regardless of methods or principles, which is $\widetilde{O}(L\sqrt{TXA})$ in [18], we have improved the computational efficiency by reducing the number of steps for model estimation and policy update from T to at most $XA\log T$ at the cost of an additional factor of $\sqrt{X}$ in the regret bound.

Notations: For any non-negative integer k, we let $[k] \triangleq \{0, 1, \cdots, k\}$. For any positive integer k, we let $[k]_+ \triangleq \{1, \cdots, k\}$. For any finite set A, $\Delta(A)$ denotes the probability simplex of A. $\mathbb{I}\{\cdot\}$ denotes the indicator function. We use Ω and O to denote the asymptotic lower and upper bounds, respectively, and use $\widetilde{O}$ to represent O by hiding the logarithmic factors. We use calligraphic and uppercase letters to denote sets and their cardinalities, respectively (e.g., $X = |\mathcal{X}|$, $X_h = |\mathcal{X}_h|$, $A = |\mathcal{A}|$).

---

[1]We have translated the bounds into our notations and model defined in Section II. Therefore, the bounds shown here differ slightly from how they appear in the original paper [16].

II. PROBLEM FORMULATION

We study an episodic discrete-time MDP defined by a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, P, r)$, where $\mathcal{X}$ is the finite state space, $\mathcal{A}$ is the finite action space, $P : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to [0, 1]$ is the transition function, and $r : \mathcal{X} \times \mathcal{A} \to [0, 1]$ is the expected reward function (shortened as reward function hereinafter). When the learner takes action $a \in \mathcal{A}$ in state $x \in \mathcal{X}$, the system transitions to state $x' \in \mathcal{X}$ with probability $P(x'|x, a)$, and the learner receives a realized reward randomly drawn from some distribution on $[0, 1]$ with mean $r(x, a)$. The process proceeds in episodes, where each episode has the same length of L time steps. The system restarts at the end of each episode, and a new episode begins with an initial state. Moreover, following the previous literature [21]–[25], we shall focus on the *layered episodic MDP* model as described in the following assumption.

Assumption 1: The episodic MDP model satisfies the following conditions:

- The state space $\mathcal{X}$ can be partitioned into $(L + 1)$ non-intersecting layers $\mathcal{X}_0, \cdots, \mathcal{X}_L$.
- $\mathcal{X}_0$ and $\mathcal{X}_L$ are singletons, i.e., $\mathcal{X}_0 = \{x_0\}$, $\mathcal{X}_L = \{x_L\}$.
- State transition takes place only between two consecutive layers, i.e., $P(x'|x, a) > 0$ only if $x \in \mathcal{X}_h$ and $x' \in \mathcal{X}_{h+1}$ for some $h \in [L - 1]$.
- The transition function P and the reward function r are time-invariant.

Remark 1: In fact, restriction to the class of layered episodic MDPs is without loss of generality because any general setting of episodic MDP can be reduced to such a layered structure [22], [23]. The reason is that non-layered structures with time-variant transition and reward functions can be reduced to layered models by creating L identical copies of the state space to construct the layers. Therefore, the algorithm and results in this paper can be applied to any general episodic MDP model up to a scaling factor that depends on the length of each episode.

The learner interacts with the environment over a fixed finite time horizon T and follows a learning algorithm to learn the optimal policy for the MDP. Only the state space $\mathcal{X}$ and the action space $\mathcal{A}$ are known to the learner ahead of time, whereas the transition function P and the reward function r remain unknown. In episode $t \in [T]_+$, the learner determines a stochastic policy $\pi_t : \mathcal{X} \to \Delta(\mathcal{A})$ where $\pi_t(a|x)$ denotes the probability of choosing a in state x and executes the policy π_t over the entire episode. In each step $h \in [L - 1]$, the learner chooses an action $a_h \sim \pi_t(\cdot|x_h)$ and receives a realized reward r_h which is randomly drawn from some unknown distribution with mean $r(x_h, a_h)$. Such information feedback is often referred to as the *bandit feedback*.

For any policy π, we define the expected reward over an episode as follows:

$$V(\pi) \triangleq \mathbb{E}\left[\sum_{h=0}^{L-1} r(x_h, a_h)\bigg| P, \pi\right]. \tag{1}$$

An optimal policy is the policy π^* which maximizes the expected reward over an episode, i.e., $\pi^* \in \mathrm{argmax}_\pi V(\pi)$. The objective of the learner is to minimize the *total regret*, which is defined as

$$\mathrm{Reg}_T = T \cdot V(\pi^*) - \sum_{t=1}^{T} V(\pi_t). \tag{2}$$

III. PRELIMINARIES: OCCUPANCY MEASURE IN EPISODIC MDP

In this section, we provide some preliminary results and analytical tools that will leverage to establish our main results. We begin with the notion of occupancy measure that was introduced in [26] and extensively used for online learning in MDPs [21]–[25].

Definition 1: Given a layered episodic MDP, the occupancy measure $q^{P,\pi} : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to [0,1]$ induced by transition function P and policy π is defined as

$$q^{P,\pi}(x,a,x') = \mathbb{P}\{x_{h(x)} = x, a_{h(x)} = a, x_{h(x)+1} = x' | P, \pi\}, \tag{3}$$

where $h(x)$ is the index of the layer to which state x belongs.

In other words, $q^{P,\pi}(x,a,x')$ represents the probability of visiting the state-action-state triple (x,a,x') in the MDP with transition function P under policy π. Next, let us define Q to be the set of all valid occupancy measures that can be induced by some arbitrary transition function P and some policy π. We have the following statement about Q.

Proposition 1 ([22], [23], [25]): Q is a non-empty polytope and has the following representation:

$$Q = \left\{ q \in \mathbb{R}_+^{XAX} \middle| \right.$$

$$\sum_{x \in \mathcal{X}_h} \sum_{a \in \mathcal{A}} \sum_{x' \in \mathcal{X}_{h+1}} q(x,a,x') = 1, \forall h \in [L-1],$$

$$\sum_{x' \in \mathcal{X}_{h-1}} \sum_{a \in \mathcal{A}} q(x',a,x) = \sum_{a \in \mathcal{A}} \sum_{x' \in \mathcal{X}_{h+1}} q(x,a,x'),$$

$$\forall x \in \mathcal{X}_h, h \in [L-1]_+,$$

$$q(x,a,x') = 0,$$

$$\left. \forall x \in \mathcal{X}_h, a \in \mathcal{A}, x' \notin \mathcal{X}_{h+1}, h \in [L-1] \right\}. \tag{4}$$

Given an episodic MDP with transition function P and policy π, the induced occupancy measure $q^{P,\pi}$ is uniquely determined by (3). Conversely, any occupancy measure $q \in Q$ can induce some transition function P^q and some policy π^q, as shown in the following proposition.

Proposition 2 ([22], [23], [25]): Any valid occupancy measure $q \in Q$ can induce a transition function P^q and a policy π^q given by

$$P^q(x'|x,a) = \frac{q(x,a,x')}{\sum_{y \in \mathcal{X}_{h(x)+1}} q(x,a,y)}, \tag{5}$$

$$\pi^q(a|x) = \frac{\sum_{x' \in \mathcal{X}_{h(x)+1}} q(x,a,x')}{\sum_{b \in \mathcal{A}} \sum_{x' \in \mathcal{X}_{h(x)+1}} q(x,b,x')}. \tag{6}$$

Moreover, when the denominators are non-zero, P^q and π^q are uniquely defined.

Given a fixed transition function P, denote by Q^P the set of all valid occupancy measures associated with the transition function P. The following proposition shows that Q^P is a polytope, which we shall refer to as the *occupancy measure polytope induced by* P.

Proposition 3 ([22], [23], [25], [26]): Q^P is a non-empty polytope and has the following representation:

$$Q^P = Q \cap$$

$$\left\{ q \in \mathbb{R}_+^{XAX} \middle| q(x,a,x') = P(x'|x,a) \sum_{y \in \mathcal{X}_{h+1}} q(x,a,y), \right.$$

$$\left. \forall x \in \mathcal{X}_h, a \in \mathcal{A}, x' \in \mathcal{X}_{h+1}, h \in [L-1] \right\}. \tag{7}$$

The occupancy measure allows us to reduce the task of learning the optimal policy to the task of learning the optimal occupancy measure over Q^P. However, Q^P itself is unknown. Therefore, a natural approach is to construct and maintain a set of plausible transition functions $\mathcal{P}$ such that with high probability, $P \in \mathcal{P}$ and $Q^P \subseteq Q^{\mathcal{P}}$, where $Q^{\mathcal{P}} = \cup_{P \in \mathcal{P}} Q^P$ is the set of all occupancy measures associated with the transition functions in $\mathcal{P}$, and solve for the optimal occupancy measure over $Q^{\mathcal{P}}$. As learning proceeds, the hope is that the size of $\mathcal{P}$ and $Q^{\mathcal{P}}$ shrinks, and eventually $Q^{\mathcal{P}}$ gets as close to Q^P as possible.

With slight abuse of notation, let us define $r(x,a,x') = r(x,a)$ for every $(x,a,x') \in \mathcal{X} \times \mathcal{A} \times \mathcal{X}$. It has been shown in [22], [23], [25] that

$$\mathbb{E}\left[\sum_{h=0}^{L-1} r(x_h, a_h) \middle| P, \pi \right] = \langle q^{P,\pi}, r \rangle. \tag{8}$$

Therefore, if we define $q^* \in \mathrm{argmax}_{q \in Q^P} \langle q, r \rangle$, the regret (2) can be represented by

$$\mathrm{Reg}_T = T \cdot \langle q^*, r \rangle - \sum_{t=1}^{T} \langle q^{P,\pi_t}, r \rangle = \sum_{t=1}^{T} \langle q^* - q^{P,\pi_t}, r \rangle. \tag{9}$$

IV. ALGORITHM

The main algorithm we propose is called "Upper Confidence Reinforcement Learning using Linear Programming" (UCRL-LP) algorithm and is presented in Algorithm 1. The proposed algorithm is mainly inspired by UCRL [15] and UCRL2 [16] algorithms which use upper confidence bounds to determine an optimistic policy and O-REPS [21], and UC-O-REPS [22] and UOB-REPS [23] algorithms which solve online convex optimization on the occupancy measure space. The algorithm consists of two main parts: maintaining confidence sets for transition and reward functions (estimation) and solving a linear program for occupancy measure update (optimization). In the following subsections, we proceed to describe in detail the main steps in Algorithm 1.

A. Confidence Sets

This idea originates from the concept of "optimism in the face of uncertainty". The technique of constructing confidence set for transition and reward functions has been widely adopted in the existing literature of online MDP and

Algorithm 1 UCRL-LP Algorithm

Input: State space $\mathcal{X}$, action space $\mathcal{A}$, time horizon T, confidence parameter δ.

Initialization:
Epoch index: $i = 1$.
Confidence sets:
   $\mathcal{P}_1$ = set of all transition functions, $Q^{\mathcal{P}_1} = Q$.
Counters, occupancy measure, and estimators $\forall (x, a, x')$:
$N_1(x,a) = N_1(x,a,x') = n_1(x,a) = n_1(x,a,x') = 0,$
$\hat{q}_1(x,a,x') = \dfrac{1}{X_h A X_{h+1}},$
$\bar{r}_1(x,a) = 0, \quad \hat{r}_1(x,a) = 1.$
Policy: $\pi_1 = \pi^{\hat{q}_1}$.

for $t = 1, \cdots, T$ **do**
   Execute policy π_i.
   Obtain trajectory $\{x_h, a_h, r_h^{(t)}\}_{h=0}^{L-1}$.
   Update counters $\forall h$:
      $n_i(x_h, a_h) \leftarrow n_i(x_h, a_h) + 1,$
      $n_i(x_h, a_h, x_{h+1}) \leftarrow n_i(x_h, a_h, x_{h+1}) + 1.$
   Update empirical estimator $\bar{r}_i \ \forall h$:
   $\bar{r}_i(x_h, a_h) \leftarrow \bar{r}_i(x_h, a_h) + \dfrac{r_h^{(t)} - \bar{r}_i(x_h, a_h)}{N_i(x_h, a_h) + n_i(x_h, a_h)}.$
   if $\exists (x, a) \in \mathcal{X} \times \mathcal{A}, n_i(x, a) \geq N_i(x, a)$ **then**
      Let $t(i)$ be the start of epoch i. Start a new epoch
         $i \leftarrow i + 1, \quad t(i) \leftarrow t + 1.$
      Update counters $\forall (x, a, x')$:
         $N_i(x,a) \leftarrow N_{i-1}(x,a) + n_{i-1}(x,a),$
      $N_i(x,a,x') \leftarrow N_{i-1}(x,a,x') + n_{i-1}(x,a,x'),$
         $n_i(x,a) \leftarrow 0, \quad n_i(x,a,x') \leftarrow 0.$
      Update empirical estimator $\bar{r}_i \ \forall (x, a)$:
         $\bar{r}_i(x,a) \leftarrow \bar{r}_{i-1}(x,a).$
      Update confidence set $\mathcal{P}_i$ based on Eq. (10)-(12).
      Update $\hat{r}_i$ based on Eq. (15)-(16).
      Update occupancy measure $\hat{q}_i$ by solving LP1.
      Update policy $\pi_i = \pi^{\hat{q}_i}$.
   end if
end for

---

reinforcement learning [15], [16], [20], [22], [23], [25]. The algorithm maintains confidence sets for both P and r such that with high probability, P and r lie in those confidence sets, and the estimate becomes more and more accurate as learning proceeds.

To construct such confidence sets, the algorithm keeps track of the number of visits to each (x, a)-pair and to each (x, a, x')-triplet and computes the empirical mean of transition and reward functions. The confidence set is constructed based on the empirical mean and an extra term that depends on the number of visits. The algorithm UCRL-LP proceeds in epochs of random length, and a doubling epoch schedule is adopted. The confidence sets for P and r and the occupancy measure are updated only at the beginning of each epoch. The first epoch starts at $t = 1$, and each epoch will end whenever the number of visits to some (x, a)-pair is doubled compared to its value at the beginning of the epoch. Such a technique is deployed in existing algorithms such as UCRL2 [16], UC-O-REPS [22] and UOB-REPS [23]. This ensures that the confidence sets and the occupancy measure to be implemented will be updated only when there are significant improvements in the confidence sets. Therefore, this technique largely improves the computational efficiency of the algorithm without negatively affecting its performance. The total number of steps for policy (occupancy measure) update is at most $XA \log T$ [16], compared to T steps of policy update from general reinforcement learning algorithms which update the policy after each episode.

Specifically, let $N_i(x, a)$ and $N_i(x, a, x')$ be the number of visits to (x, a)-pair and (x, a, x')-triplet at the beginning of epoch i. The empirical estimation of P in epoch i is

$$\bar{P}_i(x'|x, a) = \frac{N_i(x, a, x')}{\max\{1, N_i(x, a)\}}. \tag{10}$$

The confidence set $\mathcal{P}_i$ includes the transition functions that are "close enough" to $\bar{P}$ with respect to a confidence radius $\epsilon_i(x, a)$ defined by

$$\epsilon_i(x, a) = \sqrt{\frac{2X_{h(x)+1} \log(TXA/\delta)}{\max\{1, N_i(x, a)\}}}, \tag{11}$$

where $\delta \in (0, 1)$ is some confidence parameter. The confidence set $\mathcal{P}_i$ is defined by

$$\mathcal{P}_i = \{P : \|P(\cdot|x, a) - \bar{P}_i(\cdot|x, a)\|_1 \leq \epsilon_i(x, a), \\ \forall x \in \mathcal{X}, a \in \mathcal{A}\}. \tag{12}$$

The following lemma provides a closed-form description of $Q^{\mathcal{P}_i}$, which can be represented by a system of linear constraints. This allows us to update the occupancy measure efficiently by simply solving a linear program over the polytope $Q^{\mathcal{P}_i}$.

Lemma 1 (Theorem 4.2 in [22]): If $\mathcal{P}_i$ is defined by Eq. (12), $Q^{\mathcal{P}_i}$ is a non-empty polytope and has the following representation:

$$Q^{\mathcal{P}_i} = Q \cap \left\{ q \in \mathbb{R}_+^{XAX}, \epsilon \in \mathbb{R}_+^{XAX} \right|$$
$$q(x, a, x') - \bar{P}_i(x'|x, a) \sum_{y \in \mathcal{X}_{h+1}} q(x, a, y) \leq \epsilon(x, a, x'),$$
$$\forall x \in \mathcal{X}_h, a \in \mathcal{A}, x' \in \mathcal{X}_{h+1}, h \in [L-1],$$
$$\bar{P}_i(x'|x, a) \sum_{y \in \mathcal{X}_{h+1}} q(x, a, y) - q(x, a, x') \leq \epsilon(x, a, x'),$$
$$\forall x \in \mathcal{X}_h, a \in \mathcal{A}, x' \in \mathcal{X}_{h+1}, h \in [L-1],$$
$$\sum_{x' \in \mathcal{X}_{h+1}} \epsilon(x, a, x') \leq \epsilon_i(x, a) \sum_{x' \in \mathcal{X}_{h+1}} q(x, a, x'),$$
$$\left. \vphantom{\sum} \forall x \in \mathcal{X}_h, a \in \mathcal{A}, h \in [L-1] \right\}. \tag{13}$$

Next, we consider the following lemma, which states that $\mathcal{P}_i$ contains P with high probability.

Lemma 2 (Lemma 4.1 in [22]): For any $\delta \in (0,1)$, with probability at least $1 - \delta$, $P \in \mathcal{P}_i$ for all i.

Similarly, the empirical estimation of r in epoch i is

$$\bar{r}_i(x,a) = \frac{\sum_{t=1}^{t(i)-1} \mathbb{I}^{(t)}\{(x,a)\} \cdot r_h^{(t)}}{\max\{1, N_i(x,a)\}}, \quad (14)$$

where $t(i)$ denotes the starting time of epoch i, $\mathbb{I}^{(t)}\{(x,a)\}$ is the indicator function that equals 1 if the (x,a)-pair is visited in round t and 0 otherwise, and $r_h^{(t)}$ denotes the realized reward at step h in round t. The confidence radius is defined by

$$b_i(x,a) = \sqrt{\frac{2\log(TXA/\delta)}{\max\{1, N_i(x,a)\}}}. \quad (15)$$

The upper and lower confidence bounds (UCB and LCB, respectively) for r are defined by

$$\hat{r}_i(x,a) = \min\{1, \bar{r}_i(x,a) + b_i(x,a)\}, \quad (16)$$
$$\check{r}_i(x,a) = \max\{0, \bar{r}_i(x,a) - b_i(x,a)\}. \quad (17)$$

A similar lemma below states that $r(x,a)$ is bounded by $\hat{r}_i(x,a)$ and $\check{r}_i(x,a)$ with high probability. The proof of this lemma follows directly using Hoeffding inequality and union bound.

Lemma 3: For any $\delta \in (0,1)$, with probability at least $1 - 2\delta$, $\check{r}_i(x,a) \leq r(x,a) \leq \hat{r}_i(x,a)$ for all i and all $(x,a) \in \mathcal{X} \times \mathcal{A}$.

B. Linear Programming for Occupancy Measure Update

In each epoch, the algorithm updates the occupancy measure by solving a linear program. As we have seen in Section II, the reinforcement learning problem is associated with an optimization problem over some occupancy measure space. As a result of Lemma 1, such an optimization problem reduces to a linear program, which can be efficiently solved by standard linear program solvers. Moreover, the occupancy measure is updated only at the beginning of each epoch, requiring at most $XA \log T$ steps for the update.

Finally, to update occupancy measure in epoch i, the algorithm determines an optimistic occupancy measure by solving the following linear program:

$$\hat{q}_i \in \operatorname*{argmax}_{q \in Q^{\mathcal{P}_i}} \langle q, \hat{r}_i \rangle. \quad (\text{LP1})$$

The algorithm implements the policy $\pi_i \triangleq \pi^{\hat{q}_i}$ induced by $\hat{q}_i$ for entire epoch i.

V. MAIN RESULTS

In this section, we present the regret bound of UCRL-LP algorithm. The proof follows by decomposing the regret into four components and bounding each of those terms separately.

Theorem 4: With probability at least $1 - 5\delta$, UCRL-LP algorithm achieves the following regret:

$$\text{Reg}_T \leq O\left(LX\sqrt{TA\log\left(\frac{TXA}{\delta}\right)}\right). \quad (18)$$

Proof: Define $q_i \triangleq q^{P,\pi_i}$, i.e., q_i is the occupancy measure induced by the policy π_i implemented in epoch i and the true (unknown) transition function P. Note that $\hat{q}_i$ and q_i are not the same: they are induced by the same policy π_i; however, they do not necessarily induce the same transition function. Furthermore, denote by $i(t)$ the index of epoch for round $t \in [T]_+$. Following [23], the regret defined by Eq. (9) can be decomposed into the following four terms:

$$\begin{aligned}
\text{Reg}_T &= \sum_{t=1}^{T} \langle q^* - q_{i(t)}, r \rangle \\
&= \underbrace{\sum_{t=1}^{T} \langle q^*, r - \hat{r}_{i(t)} \rangle}_{\text{BIAS-I}} + \underbrace{\sum_{t=1}^{T} \langle q^* - \hat{q}_{i(t)}, \hat{r}_{i(t)} \rangle}_{\text{DIFF}} \\
&+ \underbrace{\sum_{t=1}^{T} \langle \hat{q}_{i(t)} - q_{i(t)}, \hat{r}_{i(t)} \rangle}_{\text{ERROR}} + \underbrace{\sum_{t=1}^{T} \langle q_{i(t)}, \hat{r}_{i(t)} - r \rangle}_{\text{BIAS-II}}.
\end{aligned}$$
(19)

BIAS-I and BIAS-II are due to the use of a biased estimator $\hat{r}_i$ in the algorithm. DIFF represents the difference in solutions to the linear program. ERROR measures the error in estimating P. Each term will be analyzed separately.

Firstly, as a result of Lemma 3, with probability at least $1 - 2\delta$, $\hat{r}_{i(t)}(x,a)$ is an upper bound of $r(x,a)$ for all $t \in [T]_+$ and all $(x,a) \in \mathcal{X} \times \mathcal{A}$. Therefore, BIAS-I ≤ 0 with probability at least $1 - 2\delta$. Then, as a result of Lemma 2, with probability at least $1 - \delta$, $Q^P \subseteq Q^{\mathcal{P}_{i(t)}}$, and hence, $q^* \in Q^{\mathcal{P}_{i(t)}}$ for all $t \in [T]_+$. By the definition of $\hat{q}_{i(t)}$ in LP1 and by optimality, with probability at least $1 - \delta$, DIFF ≤ 0. Next, we proceed to bound the remaining terms ERROR and BIAS-II.

Bounding ERROR: The term ERROR is due to the learner's lack of knowledge about transition function P and hence comes from the estimation error for P, which is associated with the confidence set $\mathcal{P}_i$. The following lemma is the key lemma for bounding ERROR, the proof of which is in [1].

Lemma 4: With probability at least $1 - 2\delta$, UCRL-LP algorithm ensures that

$$\sum_{t=1}^{T} \|\hat{q}_{i(t)} - q_{i(t)}\|_1 \leq 12LX\sqrt{TA\log\left(\frac{TXA}{\delta}\right)}. \quad (20)$$

Lemma 4 bounds the accumulated difference between occupancy measures induced by the implemented policy over the actual and the estimated transition functions. As a result, ERROR can be bounded using the following lemma, whose proof uses Lemma 4 and can be found in [1].

Lemma 5: With probability at least $1 - 2\delta$,

$$\text{ERROR} \leq O\left(LX\sqrt{TA\log\left(\frac{TXA}{\delta}\right)}\right). \quad (21)$$

Bounding BIAS-II: The term BIAS-II represents the accumulated difference between the UCB of r and its expectation. Note that Lemma 3 states that a high-confidence lower bound of r is $\check{r}_{i(t)}$ for all $t \in [T]_+$. Therefore, it suffices to

bound $\sum_{t=1}^{T}\langle q_{i(t)}, \hat{r}_{i(t)} - \tilde{r}_{i(t)}\rangle$. BIAS-II is bounded by the following lemma, the proof of which can be found in [1].

Lemma 6: With probability at least $1 - 2\delta$,

$$\text{BIAS-II} \leq O\left(\sqrt{TLXA \log\left(\frac{TXA}{\delta}\right)}\right). \quad (22)$$

Finally, combining Lemmas 5 and 6 with (19) and using the fact that BIAS-I ≤ 0 and DIFF ≤ 0 completes the proof. $\blacksquare$

VI. CONCLUSION

In this paper, we devise an efficient algorithm for reinforcement learning in episodic MDP which achieves $\widetilde{O}(LX\sqrt{TA})$ regret and requires at most $XA \log T$ times of solving a linear program for its policy updates. The algorithm maintains confidence sets and solves linear optimization on polytope of occupancy measure which can be accomplished by standard linear programming solvers. Our contributions include: i) reducing an extra factor of $\sqrt{L}$ in the regret bound compared to the earlier works that deploy confidence set framework, ii) reducing the computational complexity from $O(T)$ to $O(\log T)$ compared to the earlier works that adopt a different method but with a slightly tighter regret bound. An interesting future research direction is removing a $\sqrt{X}$ factor in the regret bound by combining the occupancy measure framework used in this work with some Q-learning-based algorithm.

REFERENCES

[1] V. Leon and S. R. Etesami, "Online reinforcement learning in Markov decision process using linear programming," *arXiv:2304.00155*, 2023.

[2] E. Altman, *Constrained Markov Decision Processes*. CRC Press, 1999, vol. 7.

[3] D. Bertsekas, *Dynamic Programming and Optimal Control: Volume I*. Athena Scientific, 2012, vol. 1.

[4] T. Gabel and M. Riedmiller, "On a successful application of multi-agent reinforcement learning to operations research benchmarks," in *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*. IEEE, 2007, pp. 68–75.

[5] K. G. Vamvoudakis, Y. Wan, F. L. Lewis, and D. Cansever, *Handbook of Reinforcement Learning and Control*. Springer, 2021.

[6] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[7] A. S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning: Applications on robotics," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 153–173, 2017.

[8] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.

[9] S. G. Khan, G. Herrmann, F. L. Lewis, T. Pipe, and C. Melhuish, "Reinforcement learning and optimal adaptive control: An overview and implementation examples," *Annual Reviews in Control*, vol. 36, no. 1, pp. 42–59, 2012.

[10] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2042–2062, 2018.

[11] H.-D. Tran, F. Cai, M. L. Diego, P. Musau, T. T. Johnson, and X. Koutsoukos, "Safety verification of cyber-physical systems with reinforcement learning control," *ACM Trans. Embed. Comput. Syst.*, vol. 18, no. 5s, oct 2019.

[12] S. R. Etesami and T. Başar, "Dynamic games in cyber-physical security: An overview," *Dynamic Games and Applications*, vol. 9, no. 4, pp. 884–913, 2019.

[13] A. Coronato, M. Naeem, G. De Pietro, and G. Paragliola, "Reinforcement learning for intelligent healthcare applications: A survey," *Artificial Intelligence in Medicine*, vol. 109, p. 101964, 2020.

[14] Z. Wang and T. Hong, "Reinforcement learning for building controls: The opportunities and challenges," *Applied Energy*, vol. 269, p. 115036, 2020.

[15] P. Auer and R. Ortner, "Logarithmic online regret bounds for undiscounted reinforcement learning," in *Advances in Neural Information Processing Systems*, B. Schölkopf, J. Platt, and T. Hoffman, Eds., vol. 19. MIT Press, 2006.

[16] T. Jaksch, R. Ortner, and P. Auer, "Near-optimal regret bounds for reinforcement learning," *J. Mach. Learn. Res.*, vol. 11, p. 1563–1600, aug 2010.

[17] P. L. Bartlett and A. 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*, ser. UAI '09. Arlington, Virginia, USA: AUAI Press, 2009, p. 35–42.

[18] M. G. Azar, I. Osband, and R. Munos, "Minimax regret bounds for reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 263–272.

[19] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan, "Is Q-learning provably efficient?" in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[20] S. Agrawal and R. Jia, "Optimistic posterior sampling for reinforcement learning: worst-case regret bounds," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[21] A. Zimin and G. Neu, "Online learning in episodic markovian decision processes by relative entropy policy search," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013.

[22] A. Rosenberg and Y. Mansour, "Online convex optimization in adversarial Markov decision processes," in *International Conference on Machine Learning*, 2019.

[23] C. Jin, T. Jin, H. Luo, S. Sra, and T. Yu, "Learning adversarial Markov decision processes with bandit feedback and unknown transition," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 4860–4869.

[24] G. Neu, A. Gyorgy, and C. Szepesvari, "The adversarial stochastic shortest path problem with unknown transition probabilities," in *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, N. D. Lawrence and M. Girolami, Eds., vol. 22. La Palma, Canary Islands: PMLR, 21–23 Apr 2012, pp. 805–813.

[25] A. Rosenberg and Y. Mansour, "Online stochastic shortest path with bandit feedback and unknown transition function," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[26] E. Altman, *Constrained Markov decision processes: stochastic modeling*. Routledge, 1999.