Online Reinforcement Learning in Periodic MDP
Ayush Aniket and Arpan Chattopadhyay, Member, IEEE

Abstract—We study learning in periodic Markov decision process (MDP), a special type of nonstationary MDP where both the state transition probabilities and reward functions vary periodically, under the average reward maximization setting. We formulate the problem as a stationary MDP by augmenting the state space with the period index and propose a periodic upper confidence bound reinforcement learning-2 (PUCRL2) algorithm. We show that the regret of PUCRL2 varies linearly with the period N and as $O(\sqrt{T \log T})$ with the horizon length T. Utilizing the information about the sparsity of transition matrix of augmented MDP, we propose another algorithm [periodic upper confidence reinforcement learning with Bernstein bounds (PUCRLB)] which enhances upon PUCRL2, both in terms of regret ($O(\sqrt{N})$ dependency on period] and empirical performance. Finally, we propose two other algorithms U-PUCRL2 and U-PUCRLB for extended uncertainty in the environment in which the period is unknown but a set of candidate periods are known. Numerical results demonstrate the efficacy of all the algorithms.

Impact Statement—The applications of nonstationary reinforcement learning (RL), periodic Markov decision processes (MDPs).

NOMENCLATURE

| Symbol | Description |
|--------|-------------|
| $S$    | Set of Finite State Space |
| $A$    | Set of finite action space |
| $N$    | Period of the PMDP |
| $n$    | Period index of the PMDP |
| $k$    | Episode index |
| $v_k((s, n), a)$ | Counter for $((s, n), a)$ visit in the $k$th episode |
| $n_k((s, n), a)$ | Counter for $((s, n), a)$ visit till the $k$th episode |
| $n_k((s, n), a, s')$ | Counter for $((s, n), a, s')$ visit till the $k$th episode |
| $\hat{p}_k((s, n), a)$ | Estimated transition probability for $((s, n), a, s')$ in the $k$th episode |
| $\hat{r}_k((s, n), a)$ | Estimated reward for $((s, n), a)$ in the $k$th episode |
| $\hat{\pi}_k$ | Estimated policy in the $k$th episode |
| $\hat{\sigma}^2_{p,k}(s'(n), a)$ | Estimated population variance for transition probability for $((s, n), a, s')$ in the $k$th episode |
| $\hat{\sigma}^2_{r,k}(s'(n), a)$ | Estimated population variance for reward for $((s, n), a)$ in the $k$th episode |

I. INTRODUCTION

REINFORCEMENT learning (RL) deals with the problem of optimal sequential decision making in an unknown environment. Sequential decision making in an environment with an unknown statistical model is typically modeled as a
Markov decision process (MDP) where the decision maker, at each time step \( t \), has to take an action \( a_t \) based on the state \( s_t \) of the environment, resulting in a probabilistic transition to the next state \( s_{t+1} \) and a reward \( r_t \) accrued by the decision maker depending on the current state and current action. RL has applications in many areas including robotics [1], resource allocation in wireless networks [2], and finance [3].

In a stationary MDP, the unknown transition probabilities and reward functions are invariant with time. However, the ubiquitous presence of nonstationarity in real-world scenarios often limits the application of stationary RL algorithms. Most of the existing works in nonstationary RL require information about the maximum possible amount of changes that occur in the environment via variation budget in the transition and reward function, or via the number of times the environment changes; this does not require any assumption on the nature of nonstationarity in the environment. On the contrary, we consider a periodic MDP whose state transition probabilities and reward functions are unknown but periodic with a known period \( N \). A motivating example for such a PMDP would be a scheduling operation for an industrial application that has diurnal variation. Each hour of a day can be viewed as a period index, and it repeats in 24 h. Clearly, the transition kernel on a hourly time-scale will be periodic with 24-h period. The period index will be augmented with the original state information; we refer to this process as state augmentation, and it is explained in detail in further sections. In such setting, we propose PUCRL2 and PUCRLB algorithms and analyze their regret. Also, for a setting in which the period is unknown, we propose two other algorithms U-PUCRL2 and U-PUCRLB and demonstrate their performance via simulation.

**Related Work**: Nonstationary RL has been extensively studied in a variety of scenarios [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. Auer et al. [4] propose a restart version of the popular UCRL2 algorithm meant for stationary RL problems, which achieves an \( O(T^{1/3}T^2/3) \) regret (where \( T \) is the number of time steps), under the setting in which the MDP changes at most \( l \) number of times. In the same setting, Gajane et al. [5] show that UCRL2 with sliding windows achieves the same regret. In time-varying environment, a more apposite measure for performance of an algorithm is dynamic regret which measures the difference between accumulated reward through online policy and that of the optimal offline nonstationary policy. This was first analyzed in [6] in a solely reward varying environment. Ortner et al. [7] propose first variational dynamic regret bound of \( O(V^{1/3}T^2/3) \), where \( V \) represents the total variation in the MDP. The work of [8] provides the sliding-window UCRL2 with confidence widening, which achieves an \( O((B_r + B_p)^{1/4}T^{3/4}) \) dynamic regret, where \( B_r \) and \( B_p \) represent the maximum amount of possible variation in reward function and transition kernel, respectively. They also propose a Bandit-over-RL (BORL) algorithm which tunes the UCRL2-based algorithm in the setting of unknown variational budgets. Further, in the model-free and episodic setting, Wei and Luo [14] propose policy optimization algorithms and Fei et al. [9] propose RestartQ-UCB that achieves a dynamic regret bound of \( O(\Delta^{1/3}HT^2/3) \), where \( \Delta \) represents the amount of changes in the MDP and \( H \) represents the episode length. The article [10] studies a kernel-based approach for nonstationarity in MDPs with metric spaces. In the linear MDP case, Mao et al. [11] and Zhou et al. [12] provide optimal regret guarantees. Finally, Wei and Luo [14] provide a black-box algorithm which turns any (near-)stationary algorithm to work in a nonstationary environment with optimal dynamic regret \( \tilde{O}(\min\{\sqrt{T}, \Delta^{1/3}T^{2/3}\}) \), where \( L \) and \( \Delta \) represent the number and amount of changes of the environment, respectively.

Periodic MDP (PMDP) has been marginally studied in the literature. Riis [15] studies it in the discounted reward setting, where a policy-iteration algorithm is proposed. Veugen et al. [16] propose the first state-augmentation method for conversion of PMDP into a stationary one and analyze the performance of various iterative methods for finding the optimal policy. Recently, Hu and Defourny [17] derive a corresponding value iteration algorithm suitable for periodic problems in discounted reward case and provide near-optimal bounds for greedy periodic policies. To our knowledge, RL in PMDP has not been studied.

In this article, we make the following contributions.

1) In Section III, we study a special form of nonstationarity where the unknown reward and transition functions vary periodically with a known period \( N \). We propose a modification PUCRL2 of UCRL2, which treats the periodic MDP as stationary MDP with augmented state space. We derive a static regret bound which has a linear dependence on \( N \) and sublinear dependence on \( T \).

2) By utilizing the information about the sparsity of the transition matrix of augmented MDP, we propose another algorithm PUCRLB, a variant of UCRLB. PUCRLB achieves a better regret bound than PUCRL2; its regret has a \( \sqrt{N} \) dependence on period (Section III).

3) Further, in Section IV, we study an extended uncertainty environment wherein the period information is unknown and hidden among a set of candidate periods. We propose two algorithms U-PUCRL2 and U-PUCRLB and demonstrate their performance numerically in Section V.

**II. PROBLEM FORMULATION**

We consider a discrete time PMDP with a finite state space \( \mathcal{S} \) where \(|\mathcal{S}| = S\), a finite action space \( \mathcal{A} \) where \(|\mathcal{A}| = A \). \( N \geq 2 \) is an integer value representing the period of the PMDP. \( p_n(s'|s,a) \in \mathcal{S} \times \mathcal{A} \times \mathcal{S} \) is the probability for the next state given current state-action pair, and \( r_n(s,a) \in \mathcal{S} \times \mathcal{A} \) is the mean reward given current state-action pair, for all period indices \( n \in \{1, 2, ..., N\} \).

Let us define \( p_0(s,a) \) as the transition probability vector for a given \( s,a \) pair at time \( t \). By the periodicity assumption, \( p_{t+N}(s,a) = p_t(s,a) \) and \( r_{t+N}(s,a) = r_t(s,a) \in \mathcal{S} \times \mathcal{A}, \forall t \geq 1 \). The time horizon length is \( T >> N \).

Now, the PMDP can be transformed into a stationary MDP with augmented state-space (henceforth referred as AMDP). In this AMDP, we couple the period index and states together to obtain an augmented state space \( \mathcal{S}' = \mathcal{S} \times \{1, 2, ..., N\} \); if the
state of the original MDP is \( s \) at time \( t \), then the corresponding state in the AMDP will be \((s, (t-1) \mod N) + 1\), where \( \mod \) represents the modulo operator. Consequently, the (time-homogeneous) transition probability of the AMDP for current state \( s \) and current action \( a \) becomes the following:

\[
p'(s', n')|s, n, a) = \begin{cases} 0, & n' \neq n + 1 \mod N \\ p_n(s'|s, a), & n' = n + 1 \mod N. \end{cases}
\]

The corresponding mean reward of the AMDP is given by \( r'(s, n, a) = r_n(s, a) \). The probability mass function of the next augmented state given current (state, period)-action pair is denoted by \( p'((s, n), a) \). Obviously, under any deterministic stationary policy for the AMDP, each (state, period index) pair can only be visited after \( N \) number of time steps. Thus, the PMDP becomes a stationary AMDP with a periodic transition matrix as depicted in Fig. 1. Let \( \rho^* \) denote the optimal time-averaged (average expected reward over large number of time steps and then taking a Cesaro limit) reward [18, Sec. 8.2.1] of the AMDP. In this article, we seek to develop an RL algorithm to minimize the static regret with respect to this optimal average reward \( \rho^* \). Let \( \pi \) be any generic policy for the AMDP. Our problem is to minimize the expected static regret over all policies

\[
\min_{\pi} \sum_{t=1}^{T} \left( \rho^* - \mathbb{E}_{\pi}(r_t(s_t, n_t, a_t)) \right). \tag{1}
\]

III. ALGORITHMS FOR KNOWN PERIOD

In this section, we propose two algorithms named PUCRL2 and PUCRLB for PMDP with known \( N \). While PUCRL2 is motivated by UCRL2 algorithm, the PUCRLB algorithm is developed to handle the sparsity coming from the state-augmentation operation.

A. PUCRL2 Algorithm

PUCRL2 (Algorithm 1) estimates the mean reward and the transition kernel for each augmented state-action pair, while keeping in mind that the transition occurs only to augmented states with the next period index and the probability of transitioning to other augmented states is zero. Hence, the algorithm only estimates the nonzero transition probabilities \( \hat{p}_k((s', s, a)) \) at the beginning of episode \( k \).

At each time index, PUCRL2 checks the number of hits to (state, period index, action) tuples and state transitions. Like UCRL2, PUCRL2 proceeds in episodes. The key steps are as follows.

1) **Step 1**: At the beginning of each episode, it computes the estimates of the reward function and the transition probabilities from past observations.

2) **Step 2**: With high probability, the true AMDP lies within a confidence region computed around these estimates as shown in Lemma 1.

### Algorithm 1 PUCRL2

**Input**: \( S, A, N \), confidence parameter \( \delta \in (0, 1) \).

**Initialization**: \( t = 1, n = 1 \)

for episode \( k = 1, 2, \ldots \) do

1. **Initialize episode** \( k \): For all \( (s, n), a, s' \in S \times N \times A \times S \)

   \[
   v_k((s, n), a) = 0,
   \]

   \[
   n_k((s, n), a) = \max\{1, \sum_{t=1}^{t-1} \mathbf{1}\{(s_t, n_t, a_t) = (s(n), a)\}\},
   \]

   \[
   \rho_k((s, n), a, s') = \max\{1, \sum_{t=1}^{t-1} \mathbf{1}\{(s_t, n_t, a_t, s_{t+1}) = (s(n), a, s')\}\}
   \]

2. **Update the confidence set**: We define the confidence region for transition probability and reward functions as:

   \[
   \mathcal{P}((s, n), a) = \{\hat{p}((s, n), a) : ||\hat{p}((s, n), a) - \hat{p}_k((s, n), a)||_1 \leq \sqrt{\frac{14SN \log(2A_N/\delta)}{\rho_k((s, n), a)}} \}
   \]

   \[
   \mathcal{R}((s, n), a) = \{\hat{r}((s, n), a) : ||\hat{r}((s, n), a) - \hat{r}_k((s, n), a)||_1 \leq \sqrt{\frac{7 \log(2A_N/\delta)}{2n_k((s, n), a)}} \}
   \]

Then, \( \mathcal{M}_k \) is the set of all AMDP models, such that (4) and (5) is satisfied for all \((s, n), a\) pair.

3. **Optimistic Planning**: Compute \((\hat{M}_k, \hat{\pi}_k) = \text{Modified-Extended Value Iteration} 2(\mathcal{M}_k, \epsilon_k = 1/\sqrt{t_k})\)

4. **Execute Policies**:

   while \( v_k((s, n), a) < n_k((s, n), a) \) do

   Draw \( a_t \sim \hat{\pi}_k \); observe reward \( r_t \) and next state \( s_{t+1} \).

   Set \( v_k((s_t, n_t), a_t) = \hat{v}_k((s_t, n_t), a_t) + 1 \) and \( t = t + 1, n = ((t-1) \mod N) + 1 \)

end while

**end for**
Algorithm 2 Modified - EVI

Input: \( \mathcal{M}_k, \epsilon_k, \tau \)
Initialization: \( u_0(s, n) = 0 \forall s, n, s^* \in \mathcal{S}, n^* \in \{1, \ldots, N\} \)
for \( i = 0, 1, 2, \ldots \) do
  \[
  u_{i+1}(s, n) = \max_{a \in A} \left\{ \max_{\forall \pi \in \mathcal{R}_i(s, n, a)} \hat{r}(s, n, a) + \tau \epsilon \right\}
  \]
  \[
  + \left( 1 - \tau \epsilon \right) u_i(s, n)
  \]
  \[
  \text{if } \max_{s, n} \left\{ u_{i+1}(s, n) - u_i(s, n) \right\} - \min_{s, n} \{ u_{i+1}(n, s) - u_i(n, s) \} \leq \epsilon \text{ then}
  \]
  Break the for loop.
end for

3) Step 3: Then, PUCRL2 utilizes the confidence bounds as in (4) and (5), to find an optimistic AMDP \( \hat{M}_k \) and policy \( \hat{\pi}_k \) using modified-EVI Algorithm 2 adapted from the extended value iteration (EVI [4, Sec. 3.1.2]).

4) Step 4: This policy \( \hat{\pi}_k \) is used to take action in the episode until the cumulative number of visits to any (state, period index) pair, stored in \( \nu_k(s, n, a) \), gets doubled; this is similar to the doubling criteria for episode termination of [4].

B. Modified-EVI

Extended value iteration is used in the class of UCRL algorithms to obtain an optimistic AMDP model and policy from a high probability confidence region. According to the convergence criteria of Extended Value Iteration as in [4, Sec. 3.1.3], aperiodicity is essential, i.e., the algorithm should not choose a policy with periodic transition matrix. However, the AMDP has a specific structure due to the periodicity of the original PMDP. Hence, in order to guarantee convergence, we modify the EVI algorithm by applying an aperiodic transformation as in [18, Sec. 8.5.4] (6). At each iteration, modified-EVI (Algorithm 2) applies a self transition probability of \( (1 - \tau) \), where \( 0 < \tau < 1 \), to the same (state, period index) pair. As shown in [18, Proposition 8.5.8], this transformation does not affect the average reward of any stationary policy.

C. Analysis

Let \( M \) be a generic AMDP designated by the transition probabilities and reward functions. Let \( T((s', n')|(s, 1)) \) denote the expected first hitting time of \( (s', n') \) for \( M \), starting from \( (s, 1) \) under a stationary policy \( \pi : \mathcal{S} \times \{1, 2, \ldots, N\} \rightarrow A \). As in [4, Definition 1], the diameter of an AMDP \( M \) is defined as

\[
D_{\text{aug}} = \max_{(s', n') \neq (s, 1), (s', s) \in \mathcal{S}^2} \min_{\pi} \mathbb{E}[T((s', n')|(M, \pi, (s, n))].
\]

Theorem 1: With probability at least \( 1 - \delta \), the regret for PUCRL2 is as follows:

\[
\Delta(\text{PUCRL2}) \leq 34D_{\text{aug}}SN \sqrt{AT \log \frac{T}{\delta}}.
\]

Proof: See Appendix A.

D. PUCRLB Algorithm

In this section, we improve upon the previous algorithm by taking into account the special structure that arises out of augmentation of PMDP. Utilizing the information about the periodicity of the transition matrix of the AMDP as discussed in Section II, we provide a modification of UCRLB algorithm, periodic upper confidence reinforcement learning with Bernstein bounds (PUCRLB). Similar to [19, Sec. 3.4], we define the following:

\[
\Gamma_S((s, n), a) = \|p_\cdot((s, n), a)\|_0 = \sum_{s'} I\{p(s'|s, n, a) > 0\}. \tag{8}
\]

Due to the periodic nature, the transition from any state-action pair \( (s, n), a \in \mathcal{S} \times \mathcal{N} \times A \) limited to \( s' \in \mathcal{S} \), where the next period index is implicit by the previous one. This speciality is highlighted upon the superscript in (8).

The PUCRLB algorithm is similar to PUCRL2. The main difference lies in the use of concentration inequalities which govern the construction of the set of candidate AMDP’s. While PUCRL2 uses Weisserman’s [20] and Hoefding’s inequalities to bound the \( L_1 \) norm of transition probability vector and reward function respectively, PUCRLB uses Empirical Bernstein Inequality [21, Th. 1] to bound the functions (step 2). The transition function is bound individually for each \( ((s, n), a, s') \) pair, where \( s' \) is an implicit representation of \( (s', (n + 1) \mod N) \). Thus, in the algorithm additionally we calculate the population variances of reward and transition probabilities estimates as

\[
\hat{\sigma}^2_{p,k}(s',(s, n),a) = \hat{p}_k(s'|s, n, a)(1 - \hat{p}_k(s'|s, n, a)) \tag{9}
\]

\[
\hat{\sigma}^2_{r,k}(s,(s, n),a) = \frac{\sum_{\tau=1}^{N} I\{(s, n+\tau),a \} = (s, n, a)}{n_k((s, n), a)} \Gamma^2_{\tau} - (\hat{r}_k((s, n), a))^2. \tag{10}
\]

Algorithm 3 details all the changes in steps 2 and 3 of PUCRL2 that yield PUCRLB.

Theorem 2: With probability at least \( 1 - \delta \), the regret for PUCRLB is as follows:

\[
\Delta(\text{PUCRLB}) \leq \beta D_{\text{aug}} S \sqrt{NAT \log \frac{T}{\delta}} \quad \text{\( \pm \Delta_1 \)}
\]

\[
+ D_{\text{aug}}S^2NA \log \frac{T}{\delta} \log (T) \quad \text{\( \pm \Delta_2 \)}.
\]

Proof: See Appendix B.
Algorithm 3 PUCRLB (Modified Step-2.3 from PUCRL2)

2. **Update the confidence set:** We define the confidence region for the transition probability function and reward functions as:

\[ B_{p,k}((s, n, a, a')) = \{ \tilde{p}_k((s, n, a), a') - \beta_{p,k}^{(s,n),a,a'}, \tilde{p}_k((s, n, a) + \beta_{p,k}^{(s,n),a,a'}) \cap [0, 1] \} \]

\[ B_{r,k}((s, n, a)) = \{ \tilde{r}_k((s, n, a), a) - \beta_{r,k}^{(s,n),a}, \tilde{r}_k((s, n, a) + \beta_{r,k}^{(s,n),a}) \cap [0, 1] \} \]

where

\[ \beta_{p,k}^{(s,n),a,a'} = \frac{2\sigma^2_{p,k}(s',(s, n, a), a')}{\log(6SN An_k/\delta)} \frac{\log(6SN An_k/\delta)}{n_k((s, n), a)} \]

\[ \beta_{r,k}^{(s,n),a} = \frac{2\sigma^2_{r,k}(s',(s, n, a), a)}{\log(6SN An_k/\delta)} \frac{\log(6SN An_k/\delta)}{n_k((s, n), a)} \]

Let \( \mathcal{M}_k \) be the set of all AMDP models coming from the confidence sets defined in (11) and (12).

3. **Optimistic Planning:** Compute \((\tilde{M}_k, \tilde{\pi}_k) = \text{Modified-Extended Value Iteration} 2(\mathcal{M}_k, \epsilon_k = 1/t_k)\)

---

**E. Complexity Analysis of PUCRL2 and PUCRLB**

In each episode, in steps 1 and 2, we initialize and compute the estimates and confidence set, which has \(O(1)\) computational cost.

In step 3, we compute the optimistic policy and as discussed in [4, Sec. 3.1.2], the computational complexity of (6) in Algorithm 2 turns out to be \((N^2 S^2 A)\) steps due to state augmentation. The number of iterations required by Algorithm 2 till convergence is not known.

Step 4 utilizes the already computed optimistic policy to take actions and update estimate counters, hence it also has \(O(1)\) computational cost.

**F. Comparison Between PUCRL2 and PUCRLB**

We compare the regret bound obtained in Theorems 1 and 2 in terms of \(O\) (i.e., ignoring logarithmic terms). For \( T \geq D_{\text{aug}} S^2 N A \),

\[ \Delta(\text{PUCRL2}) = \tilde{O}(D_{\text{aug}} S N \sqrt{AT}) \geq \tilde{O}(S \sqrt{D_{\text{aug}} N AT}) \geq \tilde{O}(D_{\text{aug}} S^2 N A) = \Delta_2. \]

Now, trivially

\[ \Delta(\text{PUCRLB}) = \tilde{O}(D_{\text{aug}} S N \sqrt{AT}) \geq \tilde{O}(D_{\text{aug}} S N \sqrt{AT}) = \Delta_1. \]

Thus, PUCRLB yields a better regret bound than PUCRL2.

---

Algorithm 4 U-PUCRL2

**Input:** \( S, A, \) confidence parameter \( \delta \in (0, 1) \), set of candidate episodes \( \mathcal{N}' = \{ N_1, N_2, N_3, ... N_t \} \)

**Initialization:** \( t = 1, \rho_0, i = 0, n_i = 1 \) where, \( n_i \in \{ 1, 2, ..., N_i \} \forall i \in \{ 1, 2, ..., l \} \)

\( \hat{p}_{i,1}((s', (s, n, a), a')) = 0, \tilde{r}_{i,1}((s, n, a), a) = 0 \)

for all \((s, n, a), s', n_i \in \{ 1, 2, ..., N_i \}, i \in \{ 1, 2, ..., l \} \)

**for episode k = 1, 2, ... do**

\( t_k = t \) (starting time of episode k)

1. **Initialize episode k:**

For all \((s, n_i), a, s' \in S \times N_i \times A \times S\) and \( i, \) where \( n_i \in \{ 1, 2, ..., N_i \}, i \in \{ 1, 2, ..., N_i \} \)

\[ n_k((s, n_i), a) = \max\{1, \sum_{\tau=1}^{t-1} I\left( ((s', n_{i-1}), a, s_{\tau+1}) = ((s, n_i), a')) \} \]

\[ \tilde{p}_{k,i}((s, n_i), a) = \frac{n_k((s, n_i), a)}{n_k((s, n_i), a)} \]

\[ \tilde{r}_{k,i}((s, n_i), a) = \frac{\sum_{\tau=1}^{t-1} (r_{\tau} I((s, n_{i-1}), a, s_{\tau+1}) = ((s, n_i), a))}{n_k((s, n_i), a)} \]

2. **Calculate estimated average reward:**

\[ \hat{p}_{k,i} = \tilde{p}_{k-1,i} + \text{Value Iteration}(\tilde{p}_{k,i}, \tilde{r}_{k,i}) \]

3. **Choose the period with highest value:**

\[ I_k = \arg \max I_k, \tilde{p}_{k,i} \]

4. **Update the confidence set:** We define the confidence region for transition probability function and reward functions as:

\[ \mathcal{P}((s, n_i), a) = \{ \tilde{p}(\cdot | (s, n_i), a) : \| \tilde{p}(\cdot | (s, n_i), a) - \tilde{p}_{k-1,i}(\cdot | (s, n_i), a) \|_1 \leq \sqrt{\frac{145N_k \log(2At_k/\delta)}{n_k((s, n_i), a)}} \} \]

\[ \mathcal{R}((s, n_i), a) = \{ \tilde{r}(\cdot | (s, n_i), a) : \| \tilde{r}(\cdot | (s, n_i), a) - \tilde{r}_{k-1,i}((s, n_i), a) \| \leq \sqrt{\frac{7 \log(2SNAt_k/\delta)}{2n_k((s, n_i), a)}} \} \]

Then, \( \mathcal{M}_{k, I_k} \) is the set of all AMDP models, such that above equations are satisfied for all \((s, n_i, a)\) tuples for all \( n_i \in \{ 1, 2, ..., N_i \} \).

5. **Optimistic Planning:** Compute \((\tilde{M}_{k, I_k}, \tilde{\pi}_{k, I_k}) = \text{Modified-EVI} 2(\mathcal{M}_{k, I_k}, \epsilon_k = 1/\sqrt{t_k})\)

6. **Execute Policies:**

while \( v_k((s, n_i), a_t) < n_k((s_t, n_t), a_t) \) do

Draw \( a_t \) according to \( \tilde{\pi}_{k, I_k} \), observe reward \( r_t \) and the next state \( s_{t+1} \).

Set \( v_k(i, (s, n_i), a_t) = v_k(i, (s, n_i), a_t) + 1 \) and \( t = t + 1, n_i = (t - 1) \mod N_i + 1 \) \( \forall i \in \{ 1, 2, ..., N_i \} \)

end while
IV. EXTENDED UNCERTAINTY: UNKNOWN PERIOD

In this section, we consider the scenario where $N$ is unknown. However, we assume a set of candidate periods $\mathcal{N} \equiv \{N_1, N_2, N_3, \ldots, N_l\}$ which contains the true period $N$. This setup demands extra exploration from the agent to identify the true period with high accuracy which can be used to then model the environment and perform exploitation.

A. U-PUCRL2 Algorithm

We provide an alternative algorithm unknown-PUCRL2 or U-PUCRL2 (Algorithm 4) for learning, which is an extension of PUCRL2. The steps are as follows.

1) Step 1: The reward function $r_{k,i}((s, n_i), a)$ and transition function $\hat{p}_{k,i}(s'|s, n_i), a)$ estimates as in (16) and (17) are maintained for each candidate period (denoted by subscript $i$) separately considering their period information is true and using it to calculate respective period indices at each time step.

2) Step 2: At the beginning of each episode $k$, these estimates are used to calculate an estimate of average reward through Value Iteration algorithm [18, Algorithm 8.5.1] modified for PMDP, for each candidate period $N_i, i \in [l]$.

3) Step 3: Based on the hypothesis that the true candidate period will have the true representation of the underlying AMDP and hence will have the highest average reward, the candidate period with the highest cumulative average reward is selected as the true period for that episode as shown in Fig. 2.

4) Step 4: The transition and reward function estimates for the true period $I_k$ are used to construct the confidence set as in (18) and (19).

5) Step 5: Based on the selected period information, policy for that episode is calculated through Algorithm 2.

6) Step 6: The policy is used to take actions and observe reward and next state transition. The observation tuple $(s_t, a_t, r_t, s_{t+1})$ is used to update the estimate for every candidate period $N_i, i \in [l]$. This is valid since the underlying AMDP would produce the same tuple even if some other candidate’s policy would have selected the same action in that state.

B. U-PUCRLB Algorithm

In a similar way, we can also design U-PUCRLB for unknown $N$ based on the parent algorithm PUCRLB 3 which utilizes the known sparsity in AMDP. Similar to PUCRLB, it uses empirical Bernstein inequality [21, Th. 1] for constructing the confidence set of candidate AMDP’s (step 4). Thus, in the algorithm, we additionally calculate the population variances of reward and transition probabilities estimates for each candidate period $N_i, i \in [l]$, as

$$\hat{\sigma}^2_{p,k,i}(s'|s, n_i), a) = \hat{p}_{k,i}(s'|s, n_i), a)(1 - \hat{p}_{k,i}(s'|s, n_i), a)$$

$$\hat{\sigma}^2_{r,k,i}(s, n_i), a) = \frac{1}{N_{i,k}} \sum_{k=1}^{N_{i,k}} \{(s, a, \gamma, r_{i,k})\}r^2_{i,k}((s, n_i), a) - (\hat{r}_{k,i}(s, n_i), a))^2.$$  

Algorithm 5 details of all the changes necessary in steps 4 and 5 of U-PUCRL2 that yield U-PUCRLB.

C. Complexity Analysis of U-PUCRL2 and U-PUCRLB

In each episode, steps 1, 4, and 6 have $O(1)$ computational cost as in Section III-E.

In step 2, the value iteration algorithm [18, Algorithm 8.5.1] runs for each candidate period $N_i$, wherein in each iteration the computational cost is $O(N^3 S^2 A)$ with an additional factor $O(|\mathcal{N}|)$ in the overall complexity.

In step 3, we choose the true period $N_{Ik}$ for that episode having a cost of $O(|\mathcal{N}|)$.

Hence, apart from the dependency on $S, A, N_i$, we have an extra-factor of $|\mathcal{N}|$. This can be taken care of by utilizing parallelization techniques.

V. NUMERICAL RESULTS

We compare the performance of all the aforementioned algorithms with four stationary and nonstationary RL algorithms.

A. Stationary RL Algorithms

These algorithms only utilize the knowledge of the state and action spaces and require no information about the periodic variation in the environment.

1) Upper Confidence Bound Reinforcement Learning-2 (UCRL2 [4]) is the state of the art algorithm in stationary RL setting.

2) Upper Confidence Bound Reinforcement Learning-3 (UCRL3 [22]) is a recent improvement over UCRL2, in...
the candidate periods \( N = \{2, 3, 4, 5, 6, 7\} \), among which the true period is \( N = 5 \). We can observe that the plot corresponding to \( N = 5 \) separates itself out very early in the episodes.

All the parameters and their values used to perform the empirical analysis. Two different values for period (\( N \)) and candidate period sets (\( N \)) were used separately with all other parameters in the table remaining same.

| Parameters                  | Values                                      |
|-----------------------------|---------------------------------------------|
| Period (\( N \))           | 5                                           |
| Candidate Period Sets (\( N \)) | \{2, 3, 4, 5, 6, 7\}                  |
| \( \delta \)                | 0.05                                        |
| Number of Independent Runs  | 30                                          |
| Time Horizon (\( T \))      | 100 000                                     |
| \( \beta_t \)               | 0.5 − \arctan (1/\( t/\pi \))/\( N \)      |

terms of regret and empirical performance. It explores the sparsity of the transition function to get a better performance.

3) **Posterior Sampling for Reinforcement Learning (PSRL [23])** is an adaption of Thompson Sampling for model-based RL. It provides state of the art guarantees in several important settings.

**B. Nonstationary RL Algorithms**

1) **Bandit-Over-Reinforcement Learning (BORL [8])** algorithm does not require the knowledge of variation budget, which distinguishes it from many other popular nonstationary RL algorithms. Consequently, BORL serves as an ideal benchmark against which we can numerically compare the performance of our algorithms.

**A. Our Experimental Setup**

We perform empirical analysis on synthetic dataset. We consider a MDP with two states \( \{s_1, s_2\} \), two actions \( \{a_1, a_2\} \).

All of the parameters used in the experiment are as in Table I. The variation in the rewards and transition function are modeled using saw-tooth functions as in Tables II and III. The results of the algorithms are compared after averaging over 30 independent runs.

---

**TABLE I**

All the parameters and their values used to perform the empirical analysis. Two different values for period (\( N \)) and candidate period sets (\( N \)) were used separately with all other parameters in the table remaining same.

| Parameters                  | Values                                      |
|-----------------------------|---------------------------------------------|
| Period (\( N \))           | 5                                           |
| Candidate Period Sets (\( N \)) | \{2, 3, 4, 5, 6, 7\}                  |
| \( \delta \)                | 0.05                                        |
| Number of Independent Runs  | 30                                          |
| Time Horizon (\( T \))      | 100 000                                     |
| \( \beta_t \)               | 0.5 − \arctan (1/\( t/\pi \))/\( N \)      |

**TABLE II**

Equations of modeling a saw-tooth reward function for each state-action pair depending on the period (\( N \))

| Reward Function Values          |               |
|---------------------------------|---------------|
| \( r_1(s_1, a_1) \)            | 0.5 + \arctan (1/\( \tan (\pi + (t + 0.5)/\( N \)))/\( N \) |
| \( r_2(s_1, a_2) \)            | 0.5 − \arctan (1/\( \tan (\pi + (t + 0.5)/\( N \)))/\( N \) |
| \( r_1(s_2, a_1) \)            | 0.4 + 0.8 * \( t/N − \text{floor}(0.5 + t/\( N \)) |
| \( r_2(s_2, a_2) \)            | 0.4 − 0.8 * \( t/N − \text{floor}(0.5 + t/\( N \)) |

**TABLE III**

Equations and values of modeling period (\( N \)) dependent transition function in terms of the parameter \( \beta_t \)

| Transition Function Values          |               |
|-----------------------------------|---------------|
| \( p_t(s_1|s_1, a_1) \)              | 1             |
| \( p_t(s_1|s_1, a_2) \)              | 1 − \( \beta_t \) |
| \( p_t(s_2|s_1, a_2) \)              | \( \beta_t \) |
| \( p_t(s_2|s_2, a_2) \)              | 1 − \( \beta_t \) |
B. Strengths of Our Algorithms

Reward plots represent the cumulative rewards accrued by an algorithm throughout a specified time horizon. In Fig. 3, we clearly observe that our algorithms outperform competing algorithms. Specifically, PUCRLB performs the best as discussed in Section III-F. We also notice that U-PUCRL2 and U-PUCRLB have similar performance because, U-PUCRL2 learns the true $N$ and then behaves like PUCRL2. The same can be observed in PUCRLB and U-PUCRLB.

Regret plots provide a visual representation of the accumulated regret, which quantifies the disparity between the optimal reward and the actual reward obtained by an algorithm throughout a defined time horizon. Obviously, an algorithm with smaller regret is always preferable. Fig. 4 shows that regret for all of our algorithms become sublinear (see Section III for discussion) and are smaller than the regrets of competing algorithms.

C. Limitations and Possible Extensions of Our Algorithms

Though PUCRL2/B do not have any evident disadvantage with respect to the competing algorithms considered here, we observe that UCRL3 and PSRL outperform UCRL2. All of these stationary algorithms require similar knowledge about the environment. This opens the possibility of designing PUCRL3 and PPSRL for PMDP, which might be able to outperform PUCRL2/B.

VI. CONCLUSION

In this article, we have studied periodic nonstationary in MDPs, where the state transition and reward functions vary periodically. Existing RL algorithms for nonstationary and stationary MDPs fail to perform optimally in this setting. We have proposed two algorithms called PUCRL2 and PUCRLB, which outperform competing algorithms. We have also extended the uncertainty in the already varying environment by considering unknown period and have shown numerically that lack of knowledge of period does not matter to the long-term reward and regret performance. However, the static regret term depends linearly on the diameter of the AMDP, the characterization of which with $N$ is still open.

APPENDIX A

PROOF OF THEOREM 1

The proof borrows some ideas from [4] and is divided into sections. In Section A of Appendix A, we upper bound the total regret by removing the randomness in the rewards accumulated. The regret in the episodes where the true AMDP does not lie in the set of plausible AMDPs is bounded above in Section B of Appendix A, and with the assumption that it does in Section C of Appendix A. Finally, we complete the proof in Section D of Appendix A.

A. Splitting into Episodes

As in [4, Sec. 4.1] using Hoeffding’s inequality, we can decompose the regret as

$$\Delta = \sum_{t=1}^{T} (\rho^* - r_t((s_t, n_t), a_t))$$

$$\leq T \rho^* - \sum_{(s,n),a} N((s,n),a) r((s,n),a) + \sqrt{\frac{5}{8} T \log \frac{8T}{\delta}}$$

with probability at least $1 - (\delta/12T^{5/4})$, where $N((s,n),a)$ is the count of (state, period)-action pair after $T$ steps.

Let there be $m$ episodes in total, thus $\sum_{k=1}^{m} v_k((s,n),a) = N((s,n),a)$.

The regret in each episode can be defined as: $\Delta_k = \sum_{(s,n),a} v_k((s,n),a)(\rho^* - r((s,n),a))$. Hence,

$$\Delta \leq m \Delta_k + \sqrt{\frac{5}{8} T \log \frac{8T}{\delta}}.$$ (24)

B. Dealing With Failing Confidence Regions

Lemma 1: For any $t \geq 1$, the probability that the true AMDP $M$ is not contained in the set of plausible AMDPs $\mathcal{M}(t)$ at time $t$ is at most $\delta/15t^6$, that is,

$$\mathbb{P}(M \notin \mathcal{M}(t)) < \delta/15t^6.$$
Proof: As in [4, Sec. C.1], we bound the transition functions using \( L^1 \)-deviation concentration inequality over \( m \) distinct events from \( l \) samples [20]

\[
P\{\| \hat{p}( \cdot ) - p( \cdot ) \|_1 \geq \epsilon_p \} \leq (2^m - 2) \exp \left( -l\epsilon_p^2/2 \right).
\]

As the state space has been augmented, we have \( SN \) states and hence \( m = SN \) events.

Thus, setting

\[
\epsilon_p = \sqrt{\frac{2}{l} \log \left( \frac{2SN20SAt^2}{\delta} \right)} \leq \sqrt{\frac{14SN \log (2At/\delta)}{l}}
\]

we get the following:

\[
P\left\{ \| \hat{p}(s, n, a) - p(s, n, a) \|_1 \geq \sqrt{\frac{14SN \log (2At/\delta)}{l}} \right\} \leq \frac{\delta}{20t^3SA}.
\]

For rewards, we use Hoeffding’s inequality to bound the deviation of empirical mean from true mean given \( l \) i.i.d samples

\[
P \{ | \hat{r} - r | \geq \epsilon_r \} \leq 2 \exp \left( -2l\epsilon_r^2 \right).
\]

Setting

\[
\epsilon_r = \sqrt{\frac{1}{2l} \log \left( \frac{120SA\tau^2}{\delta} \right)} \leq \sqrt{\frac{7}{2l} \log \left( \frac{2SA\tau}{\delta} \right)}
\]

we get for all \( (s, n, a) \) pair

\[
P\left\{ | \hat{r}(s, n, a) - r(s, n, a) | \geq \epsilon_r \right\} \leq \frac{\delta}{20t^3SA}.
\]

A union bound over all possible values of \( l \), i.e., \( l \in \{1, 2, \ldots, [t/N]\} \), gives \( n_k(s, n, a) \) denotes the number of visits in \( (s, n, a) \))

\[
P\left\{ \| \hat{p}(s, n, a) - p(s, n, a) \|_1 \geq \sqrt{\frac{14SN \log (2At/\delta)}{n_k(s, n, a)}} \right\} \leq \sum_{t=1}^{[t/N]} \frac{\delta}{20t^3SA} \leq \sum_{t=1}^{[t/N]} \frac{\delta}{20t^3SA} = \frac{\delta}{20t^6SAN}
\]

\[
P\left\{ | \hat{r}(s, n, a) - r(s, n, a) | \geq \sqrt{\frac{7 \log (2SA\tau/\delta)}{2n_k(s, n, a)}} \right\} \leq \frac{\delta}{20t^6SA}.
\]

Summing these probabilities over all (state, period)-action pairs we obtain the claimed bound \( P\{ M \notin \mathcal{M}(t) \} < \delta/15t^6 \).

Lemma 2: With probability at least \( 1 - (\delta/12T^{3/4}) \), the regret occurred due to failing confidence region, i.e.,

\[
\sum_{k=1}^{m} \Delta_k \mathbb{1}_{\{ M \notin \mathcal{M}_k \}} \leq \sqrt{T}
\]

Proof: Refer [4, Sec. 4.2] with Lemma 1 instead of [4, Appendix C.1]

C. Episodes With \( M \in \mathcal{M}_k \)

By the assumption \( M \in \mathcal{M}_k \) and [4, Th. 7], the optimistic optimal average reward of the near-optimal policy \( \tilde{\pi}_k \) chosen in modified-EVI 2 is such that \( \hat{\rho}_k \geq \rho^* - \epsilon_k \).

Thus, substituting \( \epsilon_k = 1/\sqrt{t_k} \), we can write the regret of an episode as

\[
\Delta_k = \sum_{(s, n, a)} v_k((s, n, a))(\rho^* - r((s, n, a)))
\]

\[
\leq \sum_{(s, n, a)} v_k((s, n, a))(\hat{\rho}_k - r((s, n, a)))
\]

\[
+ \sum_{(s, n, a)} v_k((s, n, a))/\sqrt{t_k}.
\]

Let us define \( i_k \) to be the last iteration when convergence criteria holds and modified-EVI terminates, thus as in [4, Sec. 4.3.1]

\[
| u_{i_k+1}(s, n) - u_{i_k}(s, n) - \hat{\rho}_k | \leq 1/\sqrt{t_k} \quad (27)
\]

for all \( (s, n) \). Expanding as in (6)

\[
u_{i_k+1}(s, n) = \tilde{r}_k((s, n), \tilde{\pi}_k(s, n))
\]

\[
+ \tau \star \left\{ \sum_{s'} u_{i_k}(s', n + 1)\hat{\rho}_k(s'|s, n, \tilde{\pi}_k(s, n)) \right\}
\]

\[
+ (1 - \tau) \star u_{i_k}(s, n).
\]

Putting it in (27), we get the following equation:

\[
\Delta_k \leq \tau \sum_{(s, n, a)} v_k((s, n, a)) \left\{ \sum_{s'} u_{i_k}(s', n + 1)\hat{\rho}_k(s'|s, n, a) - u_{i_k}(s, n) \right\}
\]

\[
+ \sum_{(s, n, a)} v_k((s, n, a))(\hat{r}_k((s, n, a)) - r((s, n, a)) \right\}
\]

\[
+ 2 \sum_{(s, n, a)} v_k((s, n, a))/\sqrt{t_k}.
\]

\[
(28)
\]
where the last inequality uses the confidence bound (4). We note that the aperiodicity transformation coefficient gets canceled out and does not appear in the regret term.

Following the proof of [4, Second term, Sec. 4.3.2], the second term in (29) can be bounded as

\[
\tau \sum_{k=1}^{m} v_k((s, n), a) \left( \sum_{s'} u_{ik}(s', n+1)p_k(s'|s, n, a) - u_{ik}(s, n) \right) \\
\leq \tau D_{aug} \sqrt{\frac{5T}{2} \log \frac{8T}{\delta} + m\tau D_{aug}} \\
\leq fD_{aug} \sqrt{\frac{5T}{2} \log \frac{8T}{\delta} + m\tau D_{aug}}/f 
\]

where the last inequality uses the confidence bound (5).

D. Completing the Proof

Thus, we can write the total episodic regret using (28), (32), (33), and (34), with probability at least \(1 - (\delta/12T^{5/4})\)

\[
\sum_{k=1}^{m} \Delta_k \mathbf{1}_{\{M \in M_k\}} \leq \sum_{k=1}^{m} \sum_{(s, n), a} v_k((s, n), a) \\
D_{aug} \left( 4SN \log \frac{2At_k/\delta}{n_k((s, n), a)} + D_{aug} \log \frac{8T}{SNA} \right) + \left( \sqrt{14} \log \frac{2SA\tau_{tk}/\delta}{2n_k((s, n), a)} \right) \\
\leq \sum_{(s, n), a} v_k((s, n), a) 2^{\tau} \sum_{(s, n), a} \sqrt{14SN \log \frac{2At_k/\delta}{n_k((s, n), a)}} D_{aug}/2^{\tau} 
\]

with probability at least \(1 - (\delta/12T^{5/4})\).
Also, noting that \( n_k((s, n), a) \leq t_k \leq T \). Thus,
\[
\sum_{k=1}^{m} \Delta_k 1_{(M \in M_k)} \leq D_{\text{aug}} \frac{\sqrt{2T \log \frac{8T}{\delta}}}{} + D_{\text{aug}} S N A \log \frac{8T}{S N A}
\]
\[
(\sqrt{2} + 1)(\sqrt{S N A T}).
\]
(35)

Using (24), (25), and (35), with probability at least 1 – \( \delta / (4T^2/\delta) \), we can bound the total regret as
\[
\Delta \leq \sum_{k=1}^{m} \Delta_k 1_{(M \in M_k)} + \sum_{k=1}^{m} \Delta_k 1_{(M \notin M_k)} + \sqrt{\frac{5}{\delta} T \log \frac{8T}{\delta}}
\]
\[
D_{\text{aug}} \frac{\sqrt{2T \log \frac{8T}{\delta}}}{} + D_{\text{aug}} S N A \log \frac{8T}{S N A}
\]
\[
+ (2D_{\text{aug}} \sqrt{14S N \log (2AT/\delta)} + 2)(\sqrt{2} + 1)(\sqrt{S N A T})
\]
\[
+ \sqrt{T} + \sqrt{\frac{5}{\delta} T \log \frac{8T}{\delta}}.
\]

Further simplifications as in [4, Appendix C.4] yield the total regret as
\[
\Delta \leq 34D_{\text{aug}} SN \sqrt{AT \log (T/\delta)}
\]
with probability at least 1 – \[ \sum_{T=2}^{\infty} (\delta / 4T^2/\delta) > 1 - \delta \) by union over all values of \( T \).

**APPENDIX B**

**PROOF OF THEOREM 2**

**A. Optimism With Concentration Inequalities**

**Lemma 3:** The probability that there exists \( k \geq 1 \) such that the true AMDP \( M \) does not belong to the set of candidate AMDP’s \( M_k \) denoted by (11) and (12) is at most \( \delta / 3 \), that is,
\[
P(\exists k \geq 1 \text{ s.t. } M \notin M_k) \leq \frac{\delta}{3}.
\]

**Proof:** As in [19, Sec. 3.2.2] we bound the probability of the event \( E = \bigcup_{k=1}^{\infty} \{ M \notin M_k \} \). Throughout the proof, we use the notation \( n_k \) instead of \( n_k((s, n), a) \) for brevity. Event \( E \) is equivalent to
\[
E \subseteq \bigcup_{(s, n), a} \{ r((s, n), a) \notin B^k_p((s, n), a) \}
\]
\[
\bigcup_{s'} \{ p(s'|s, n, a) \notin B^k_p((s, n), a, s') \}
\]

\[
P(E) \leq \sum_{(s, n), a} \sum_{k=1}^{\infty} \left\{ P(r((s, n), a) \notin B^k_p((s, n), a)) + \sum_{s'} \left[ P(p(s'|s, n, a) \notin B^k_p((s, n), a, s')) \right] \right\}
\]
where \( B^k_p((s, n), a) \) and \( B^k_p((s, n), a, s') \) are as in (11) and (12), respectively.

Let’s take a four-tuple \( ((s, n), a, s', \alpha) \in S \times P \times A \times S \), we define the following:
\[
\epsilon^{(s, n), a, s', \alpha}_{\text{app}}(s, n, a) = \frac{2 \log (30S^2NAn^2_k/\delta)}{n_k} + 3 \log (30S^2NAn^2_k/\delta)
\]
\[
\epsilon^{(s, n), a}_{\text{app}}(s, n, a) = \frac{2 \log (30S^2NAn^2_k/\delta)}{n_k} + 3 \log (30S^2NAn^2_k/\delta).
\]

Since \( \epsilon^{(s, n), a, s'}_{\text{app}} \leq \beta^{(s, n), a, s'} \) and \( \epsilon^{(s, n), a}_{\text{app}} \leq \beta^{(s, n), a, s'} \), by using empirical Bernstein inequality [21, Th. 1], we can bound the probability of the events as
\[
P\{ |\hat{r}_k((s, n), a) - r((s, n), a)| \geq \beta^{(s, n), a, s'} \}
\]
\[
\leq P\{ |\hat{r}_k((s, n), a) - r((s, n), a)| \geq \epsilon^{(s, n), a}_{\text{app}} \}
\]
\[
\leq P\{ |\hat{p}_k(s'|s, n, a) - p(s'|s, n, a)| \geq \beta^{(s, n), a, s'} \}
\]
\[
\leq P\{ |\hat{p}_k(s'|s, n, a) - p(s'|s, n, a)| \geq \epsilon^{(s, n), a}_{\text{app}} \}
\]
\[
\leq \frac{\delta}{10n^2 S^2 N A}
\]

Thus,
\[
P(E) \leq \sum_{(s, n), a} \left( \sum_{k=1}^{\infty} \left( \frac{\delta}{10n^2 S^2 N A} + \sum_{s'} \frac{\delta}{10n^2 S^2 N A} \right) \right)
\]
\[
= \frac{2 \pi^2 \delta}{60} \leq \frac{\delta}{3}.
\]

**B. Splitting into Episodes**

For the stochastic process \( X_t \equiv r_t((s_t, n_t), a_t) - r((s_t, n_t), a_t) \), \( \{ X_t \}_{t=1}^{\infty} \) is a Martingale difference sequence (MDS) with \( |X_t| \leq 1 \). Using Azuma’s Inequality for MDS [4, Lemma 10], we can write the following:

\[
P\left( \sum_{t=1}^{T} X_t \geq -\sqrt{4T \log \left( \frac{4T}{\delta} \right)} \right) \leq \frac{\delta}{16T^2}.
\]

(36)

Taking a union bound for all possible values of \( T \geq 1 \), with a probability at least \( \frac{1 - \sum_{t=1}^{\infty} (\delta / 16T^2)}{1 - \frac{\delta}{3}} \), we obtain the following:

\[
\sum_{t=1}^{T} X_t \geq -\sqrt{4T \log \left( \frac{4T}{\delta} \right)}
\]
\[
\Leftrightarrow \sum_{t=1}^{T} -r_t((s_t, n_t), a_t) \leq \sum_{t=1}^{T} -r((s_t, n_t), a_t)
\]
\[
+ \sqrt{4T \log \left( \frac{4T}{\delta} \right)}.
\]
Thus, we can decompose the total regret as
\[
\Delta = \sum_{t=1}^{T} (\rho^* - r_t(s_t, n_t, a_t)) \leq \sum_{t=1}^{T} (\rho^* - r((s_t, n_t), a_t)) + 2\sqrt{T \log 4T} \frac{4}{\delta} \\
\leq \sum_{k=1}^{m} \Delta_k + 2\sqrt{T \log 4T} \frac{4}{\delta} \tag{37}
\]
with probability at least \(1 - (\delta/3)\), where \(m\) represents the total number of episodes and episodic regret \(\Delta_k = \sum_{(s, n), a} v_k((s, n), a)(\rho^* - r((s, n), a))\).

### C. Episodic Regret

As in Section C of Appendix A, with \(\epsilon_k = 1/t_k\), the episodic regret can be decomposed as
\[
\Delta_k \leq \Delta_k^p + \Delta_k^r + 2 \sum_{(s, n), a} v_k((s, n), a) \frac{t_k}{T} \tag{38}
\]

1) **Bounding \(\Delta_k^p\):** Following the same arguments as in Section C1 of Appendix A, the first term of (29), can be bounded similarly to (32) as
\[
\begin{align*}
&\tau \sum_{(s, n), a} v_k((s, n), a) \left( \sum_{s'} w_k(s', n + 1) (\hat{p}_k(s'|s, n, a) - p_k(s'|s, n, a)) \right) \\
&\leq \sum_{(s, n), a} v_k((s, n), a) (D_{aug} \| \hat{p}_k \|_{\mathcal{H}_k} - p_k(\cdot|s, n, a)) \tag{39}
\end{align*}
\]

The second term in (29) after replacing \(u_{i_k}\) with \(w_k\) can be bounded as
\[
\begin{align*}
&\tau \sum_{t=1}^{t_k-1} \left( \sum_{s'} p_k(s'|s_t, n_t, a) w_k(s', n_{t+1}) - w_k(s_t, n_t) \right) \\
&= \tau \sum_{t=1}^{t_k-1} \left( \sum_{s'} p_k(s'|s_t, n_t, a) w_k(s', n_{t+1}) - w_k(s_{t+1}, n_{t+1}) \right) + \tau \sum_{t=1}^{t_k-1} w_k(s_{t+1}, n_{t+1}) - w_k(s_t, n_t). \tag{40}
\end{align*}
\]

The last term in (40) is a telescopic sum
\[
\begin{align*}
&\tau \sum_{t=1}^{t_k-1} w_k(s_{t+1}, n_{t+1}) - w_k(s_t, n_t) \\
&= \tau \left( (w_k(s_{t_k+1}, n_{t_k+1})) - (w_k(s_{t_k+1}, n_{t_k+1})) \right) \\
&\leq \tau \text{span}(w_k) \leq D_{aug}. \tag{41}
\end{align*}
\]

Similar to (36), for the stochastic process \(X_t = \tau \sum_{s'} p_k(s'|s_t, n_t, a) w_k(s', n_{t+1}) - w_k(s_t, n_{t+1})\) with \(|X_t| \leq \text{span}(w_k) \leq D_{aug}/\tau\), under event \(E_C\), \(\forall T \geq 1\), and using Azuma’s inequality for MDS, we get the following:
\[
P\left( \sum_{t=1}^{T} X_t \geq 2D_{aug}\sqrt{T \frac{4T}{\delta}} \right) \leq \frac{\delta}{16T^2}. \tag{42}
\]

Thus, with probability at least \(1 - \sum_{T=1}^{\infty} (\delta/16T^2) > 1 - (\delta/3)\).

2) **Bounding \(\Delta_k^r\):** Similar to Section C2 of Appendix A
\[
\begin{align*}
\Delta_k^r &\leq 2 \sum_{(s, n), a} v_k((s, n), a) \beta_{r, k}^{(s, n), a} \tag{44}
\end{align*}
\]

using the confidence bound (12).

### D. Summing Over Episodes

We state a result that would be useful later.

**Lemma 4:** It holds almost surely that \(\forall k \geq 1\) and \(\forall ((s, n), a) \in \mathcal{S} \times \mathcal{N} \times \mathcal{A}\)
\[
\sum_{k=1}^{m} \frac{v_k((s, n), a)}{\sqrt{n_k((s, n), a)}} \leq 3 \sqrt{n_{m+1}((s, n), a)} \tag{43}
\]
\[
\sum_{k=1}^{m} \frac{v_k((s, n), a)}{n_k((s, n), a)} \leq 2 + 2 \log n_{m+1}((s, n), a) \tag{44}
\]

**Proof:** Refer [19, Lemma 3.6].

Under event \(E_C\), combining the results of Lemma 3, (38), (43), and (44), we can bound the total sum of episodic regret \(\forall T \geq SNA\) as
\[
\begin{align*}
\sum_{k=1}^{m} \Delta_k &\leq D_{aug} \sum_{k=1}^{m} \sum_{(s, n), a} v_k((s, n), a) \beta_{r, k}^{(s, n), a} + 2 \sum_{k=1}^{m} \sum_{(s, n), a} v_k((s, n), a) \beta_{p, k}^{(s, n), a} \\
&= \Delta_t \tag{45}
\end{align*}
\]
By using Lemma 4,\( \hat{\sum}_{k} \sum_{(s,n),a} v_k((s,n),a) \leq \sum_{k=1}^{m} \sum_{(s,n),a} v_k((s,n),a) \leq SNA(2 + 2 \log T) \). (48)

Under event \( E^C \), combining (37), (45), (46), (47), and (48), and taking a union bound, we can bound the total regret \( \forall t \geq SNA \) as

\[
\Delta \leq 6 \sqrt{\left( \sum_{(s,n),a} \Gamma_S((s,n),a) \right) T \log \left( \frac{6SNAT}{\delta} \right) + 12S^2NA \log \left( \frac{6SNAT}{\delta} \right)(1 + \log T)} + 6 \sqrt{SNAT \log \left( \frac{6SNAT}{\delta} \right)} + 12SNA \log \left( \frac{6SNAT}{\delta} \right)(1 + \log T) + 4SNA(1 + \log T) + 2D_{aug} \sqrt{T \frac{4T}{\delta} + D_{aug} \frac{8T}{SNA} + 2 \sqrt{T \log \frac{4T}{\delta}}}.
\]

(49)

with probability at least \( 1 - (2\delta/3) \).

3) Bounding \( \Delta_4 \): Using (13) and \( n_k((s,n),a) \leq T \), \( \Delta_4 \) can be bounded as

\[
\sum_{k=1}^{m} \sum_{(s,n),a} v_k((s,n),a) \sum_{s',p,k} \beta_{(s,n),a,s'} \leq 2 \sqrt{\log \left( \frac{6SNAT}{\delta} \right)}.
\]

(45)

4) Bounding \( \Delta_5 \): Using (14), \( \Delta_5 \) can be bounded as

\[
\sum_{k=1}^{m} \sum_{(s,n),a} v_k((s,n),a) \sum_{s',p,k} \beta_{(s,n),a,s'} \leq 6 \sqrt{\left( \sum_{(s,n),a} \Gamma_S((s,n),a) \right) T \log \left( \frac{6SNAT}{\delta} \right) + 12S^2NA \log \left( \frac{6SNAT}{\delta} \right)(1 + \log T)}.
\]

(46)

5) Bounding \( \Delta_6 \): Since \( t_k \geq n_k((s,n),a) \forall (s,n),a \) and using Lemma 4, \( \Delta_6 \) can be bounded as

\[
\sum_{k=1}^{m} \sum_{(s,n),a} v_k((s,n),a) \leq \sum_{k=1}^{m} \sum_{(s,n),a} v_k((s,n),a) \leq SNA(2 + 2 \log T). \]

(48)

REFERENCES

[1] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” Int. J. Robot. Res., vol. 32, no. 11, pp. 1238–1278, 2013.
[2] J. Y. Yu and S. Mannor, “Online learning in Markov decision processes with arbitrarily changing rewards and transitions,” in Proc. Int. Conf. Game Theory Netw., 2009, pp. 314–322.
[3] V. Bacoyninis, V. Glukhov, T. Jin, J. Kochems, and D. R. Song, “Idiosyncrasies and challenges of data driven learning in electronic trading,” 2018, arXiv:1811.09549.
[4] P. Auer, T. Jaksch, and R. Ortner, “Near-optimal regret bounds for reinforcement learning,” in Proc. Adv. Neural Inf. Process. Syst., vol. 21, pp. 2–3, 2008.
[5] P. Gajane, R. Ortner, and P. Auer, “A sliding-window algorithm for Markov decision processes with arbitrarily changing rewards and transitions,” 2018, arXiv:1805.10066.
[6] Y. Li and N. Li, “Online learning for Markov decision processes in nonstationary environments: A dynamic regret analysis,” in Proc. Amer. Control Conf. (ACC), Piscataway, NJ, USA: IEEE, 2019, pp. 1232–1237.
[7] R. Ortner, P. Gajane, and P. Auer, “Variational regret bounds for reinforcement learning,” in Proc. Uncertainty Artif. Intell., PMLR, 2020, pp. 81–90.
[8] W. C. Cheung, D. Simchi-Levi, and R. Zhu, “Reinforcement learning for non-stationary Markov decision processes: The blessing of (more) optimism,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 1843–1854.

[9] Y. Fei, Z. Yang, Z. Wang, and Q. Xie, “Dynamic regret of policy optimization in non-stationary environments,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 6743–6754, 2020.

[10] O. D. Domingues, P. Ménard, M. Pirotta, E. Kaufmann, and M. Valko, “A kernel-based approach to non-stationary reinforcement learning in metric spaces,” in *Proc. Int. Conf. Artif. Intell. Statist.*, PMLR, 2021, pp. 3538–3546.

[11] W. Mao, K. Zhang, R. Zhu, D. Simchi-Levi, and T. Basar, “Near-optimal model-free reinforcement learning in non-stationary episodic MDPs,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 7447–7458.

[12] H. Zhou, J. Chen, L. R. Varshney, and A. Jagmohan, “Nonstationary reinforcement learning with linear function approximation,” 2020, arXiv:2010.04244.

[13] A. Touati and P. Vincent, “Efficient learning in non-stationary linear Markov decision processes,” 2020, arXiv:2010.12870.

[14] C.-Y. Wei and H. Luo, “Non-stationary reinforcement learning without prior knowledge: An optimal black-box approach,” in *Proc. Conf. Learn. Theory*, PMLR, 2021, pp. 4300–4354.

[15] J. O. Riis, “Discounted Markov programming in a periodic process,” *Oper. Res.*, vol. 13, no. 6, pp. 929–929, 1965.

[16] L. Veugen, J. van der Wal, and J. Wessels, “The numerical exploitation of periodicity in Markov decision processes,” *Oper.-Res.-Spektrum*, vol. 5, no. 2, pp. 97–103, 1983.

[17] Y. Hu and B. Defourny, “Near-optimality bounds for greedy periodic policies with application to grid-level storage,” in *Proc. IEEE Symp. Adaptive Dynamic Program. Reinforcement Learn. (ADPRL)*, Piscataway, NJ, USA: IEEE, 2014, pp. 1–8.

[18] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2014.

[19] R. Fruit, “Exploration-exploitation dilemma in reinforcement learning under various form of prior knowledge,” Ph.D. dissertation, Univ. Lille 1, Sci. Technol., Lille, France, CRISiS UMR 9189, 2019.

[20] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. J. Weinberger, “Inequalities for the l1 deviation of the empirical distribution,” Hewlett-Packard Labs, Tech. Rep., 2003.

[21] J.-Y. Audibert, R. Munos, and C. Szepesvári, “Exploration–exploitation tradeoff using variance estimates in multi-armed bandits,” *Theor. Comput. Sci.*, vol. 410, no. 19, pp. 1876–1902, 2009.

[22] H. Bourel, O. Maillard, and M. S. Talebi, “Tightening exploration in upper confidence reinforcement learning,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 1056–1066.

[23] I. Osband, D. Russo, and B. Van Roy, “(More) efficient reinforcement learning via posterior sampling,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, pp. 3–5, 2013.