Posterior Sampling-Based Online Learning for the Stochastic Shortest Path Model

Mehdi Jafarnia-Jahromi[1] **Liyu Chen**[3] **Rahul Jain**[2,3,4] **Haipeng Luo**[3]

[1]Google DeepMind
[2]ECE Department, University of Southern California
[3]CS Department, University of Southern California
[4]USC Center for Autonomy and AI

Abstract

We consider the problem of online reinforcement learning for the Stochastic Shortest Path (SSP) problem modeled as an unknown MDP with an absorbing state. We propose `PSRL-SSP`, a simple posterior sampling-based reinforcement learning algorithm for the SSP problem. The algorithm operates in epochs. At the beginning of each epoch, a sample is drawn from the posterior distribution on the unknown model dynamics, and the optimal policy with respect to the drawn sample is followed during that epoch. An epoch completes if either the number of visits to the goal state in the current epoch exceeds that of the previous epoch, or the number of visits to any of the state-action pairs is doubled. We establish a Bayesian regret bound of $\tilde{\mathcal{O}}(B_\star S\sqrt{AK})$, where $B_\star$ is an upper bound on the expected cost of the optimal policy, S is the size of the state space, A is the size of the action space, and K is the number of episodes. The algorithm only requires the knowledge of the prior distribution, and has no hyper-parameters to tune. It is the first such posterior sampling algorithm and outperforms numerically previously proposed optimism-based algorithms.

1 INTRODUCTION

Stochastic Shortest Path (SSP) model considers the problem of an agent interacting with an environment to reach a predefined goal state while minimizing the cumulative expected cost. Unlike the finite-horizon and discounted Markov Decision Processes (MDPs), in the SSP model, the horizon of interaction between the agent and the environment depends on the agent's actions, and can possibly be unbounded (if the goal is not reached). A wide variety of goal-oriented control and reinforcement learning (RL) problems such as navigation, game playing, etc. can be formulated as SSP problems. In the RL setting, where the SSP model is unknown, the agent interacts with the environment in K *episodes*. Each episode begins at a predefined initial state and ends when the agent reaches the goal (note that it might never reach the goal). We consider the setting where the state and action spaces are finite, the cost function is known, but the transition kernel is unknown. The performance of the agent is measured through the notion of *regret*, i.e., the difference between the cumulative cost of the learning algorithm and that of the optimal policy during the K episodes.

The agent has to balance the well-known trade-off between *exploration* and *exploitation*: should the agent *explore* the environment to gain information for future decisions, or should it *exploit* the current information to minimize the cost? A general way to balance the exploration-exploitation trade-off is to use the *Optimism in the Face of Uncertainty* (OFU) principle [Lai and Robbins, 1985]. The idea is to construct a set of plausible models based on the available information, select the model associated with the minimum cost, and follow the optimal policy with respect to the selected model. This idea is widely used in the RL literature for MDPs (e.g., [Jaksch et al., 2010, Azar et al., 2017, Fruit et al., 2018, Jin et al., 2018, Wei et al., 2020, 2021]) and also for SSP models [Tarbouriech et al., 2020, Rosenberg et al., 2020, Rosenberg and Mansour, 2020, Chen and Luo, 2021, Tarbouriech et al., 2021b].

An alternative fundamental idea to encourage exploration is to use Posterior Sampling (PS) (also known as Thompson Sampling) [Thompson, 1933]. The idea is to maintain the posterior distribution on the unknown model parameters based on the available information and the prior distribution. PS algorithms usually proceed in *epochs*. In the beginning of an epoch, a model is sampled from the posterior. The actions during the epoch are then selected according to the optimal policy associated with the sampled model. PS algorithms have two main advantages over OFU-type algorithms. First, the prior knowledge of the environment can be incorporated through the prior distribution. Second, PS algorithms

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

have shown superior numerical performance on multi-armed bandit problems [Scott, 2010, Chapelle and Li, 2011], and MDPs [Osband et al., 2013, Osband and Van Roy, 2017, Ouyang et al., 2017b].

The main difficulty in designing PS algorithms is the design of the epochs. In the basic setting of bandit problems, one can simply sample at every time step [Chapelle and Li, 2011]. In finite-horizon MDPs, where the length of an episode is predetermined and fixed, the epochs and episodes coincide, i.e., the agent can sample from the posterior distribution at the beginning of each episode [Osband et al., 2013]. Moreover, a bad policy in an episode of a finite-horizon MDP only results in constant regret. However, in the general SSP model, where the length of each episode is not predetermined and can possibly be unbounded, these natural choices for the epoch do not work. This is because sticking to a bad policy in any of the episodes prevents the agent from reaching the goal and imposes infinite regret. Indeed, the agent needs to switch policies during an episode if the current policy cannot reach the goal.

In this paper, we propose `PSRL-SSP`, the first PS-based RL algorithm for the SSP model. `PSRL-SSP` starts a new epoch based on two criteria. According to the first criterion, a new epoch starts if the number of episodes within the current epoch exceeds that of the previous epoch. The second criterion is triggered when the number of visits to any state-action pair is doubled during an epoch. Intuitively speaking, in the early stages of the interaction between the agent and the environment, the second criterion triggers more often. This criterion is responsible for switching policies during an episode if the current policy cannot reach the goal. In the later stages of the interaction, the first criterion triggers more often and encourages exploration. We prove a Bayesian regret bound of $\tilde{\mathcal{O}}(B_\star S\sqrt{AK})$, where S is the number of states, A is the number of actions, K is the number of episodes, and $B_\star$ is an upper bound on the expected cost of the optimal policy.

Our regret bound is similar to that of Rosenberg et al. [2020] and has a gap of $\sqrt{S}$ with the lower bound. Note that Tarbouriech et al. [2021b], Cohen et al. [2021], Chen et al. [2021] have proposed OFU algorithms that in theory have closed this gap for minimax regret. But as we will see in Section 5, the empirical performance of our PS algorithm is much better than that of the OFU algorithms proposed therein. Our algorithm is the *first PS algorithm* for the SSP setting. And as for finite-horizon [Osband et al., 2013] and the infinite-horizon average-cost MDPs [Ouyang et al., 2017b], despite a $\sqrt{S}$ gap to the lower bound in theory, PS algorithms significantly outperform the OFU-type algorithms empirically. The $\sqrt{S}$ gap is understood to be an artifact of the analysis and it remains an open question how to bridge it via tighter analysis for PS algorithms in general.

The **main contributions** of this paper are as follows:

Algorithmic novelty: A strength of PS algorithms is that their design follows the same general template, and in the infinite-horizon setting, it essentially boils down to the design of the epochs since the rest of the algorithm is natural. This is indeed non-trivial in the SSP setting for three reasons. First, although the SSP model seems closer to the finite-horizon MDPs (as previous OFU algorithms suggest [Cohen et al., 2021]), applying the PS algorithm of the finite-horizon MDPs [Osband et al., 2013] that samples in the beginning of the episodes does not work for the SSP model, because the policy obtained for the sampled transition kernel may not be proper. Second, artificially switching to the fast policy after some time if the current policy does not reach the goal (as in Tarbouriech et al. [2020]), makes the algorithm unnecessarily complicated. Third, applying the PS algorithm of the infinite-horizon average-cost MDPs [Ouyang et al., 2017b] to the SSP setting leads to the sub-optimal regret bound of $\mathcal{O}(K^{2/3})$. We propose a simple yet effective epoch design that yields the near-optimal regret bound of $\tilde{\mathcal{O}}(B_\star S\sqrt{AK})$. Our epoch is determined based on two criteria. The first criterion encourages exploration by controlling the number of episodes in each epoch. The second criterion controls the number of visits to state-action pairs and is responsible to switch policies if the current policy is not proper.

Analytical novelty: In finite-horizon MDPs, the regret of an episode is at-most a constant proportional to the horizon. However, the variable length of the episodes in the SSP setting, imposes a significant challenge in the analysis because there is no upper-bound on the regret of a single episode, let alone K episodes. Therefore, applying direct analysis of previous posterior sampling approaches is not possible. To handle this issue, we have used the notion of "interval" (only in the analysis) to artificially limit the total cost by definition. Then, used concentration bounds, posterior-sampling property, and careful algebraic manipulation to self-bound the total cost C_M after M intervals in terms of $\sqrt{C_M}$. This allows us to show $C_M = \mathcal{O}(\sqrt{M})$ and then translate it in terms of regret. This type of analysis is inspired by Rosenberg et al. [2020] and is not common in previous PS algorithms in finite-horizon/infinite-horizon MDPs. Note that applying the optimism-based analysis of Rosenberg et al. [2020] to the PS setting imposes new challenges that are successfully handled. More specifically, the optimistic transition kernel of Rosenberg et al. [2020] is in the confidence set with high probability. However, in the PS setting, the case where the sampled transition kernel falls outside the confidence set needs to be handled separately (see e.g., how (9) is handled with any-time Bernstein inequality). Moreover, following Hoeffding-type concentration as in Ouyang et al. [2017b], yields a sub-optimal regret bound of $\mathcal{O}(K^{2/3})$. Instead, we propose a different analysis using Bernstein-type concentration inspired by the work of Rosenberg et al. [2020] to achieve $\mathcal{O}(\sqrt{K})$ regret bound (see Lemma 4.5). The new design of the epochs requires a novel analysis in Lemma 4.4 as well.

Numerical performance: Our simulations on SSP-MountainCar and two synthetic environments verify that the PSRL-SSP algorithm outperforms the optimism-based competitors significantly, with no hyper-parameter tuning.

Related Work. Posterior Sampling. The idea of PS algorithms dates back to the pioneering work of Thompson [1933]. The algorithm was ignored for several decades until recently. In the past two decades, PS algorithms have successfully been developed for various settings including multi-armed bandits Scott [2010], Chapelle and Li [2011], Kaufmann et al. [2012], Agrawal and Goyal [2012, 2013], MDPs [Strens, 2000, Osband et al., 2013, Fonteneau et al., 2013, Gopalan and Mannor, 2015, Osband and Van Roy, 2017, Kim, 2017, Ouyang et al., 2017b, Banjević and Kim, 2019], Partially Observable MDPs [Jafarnia-Jahromi et al., 2022], Stochastic Games [Jafarnia-Jahromi et al., 2021], and Linear Quadratic Control [Abeille and Lazaric, 2017, Ouyang et al., 2017a]. The reader is referred to Russo et al. [2017] for a more comprehensive literature review.

Online Learning in SSP. Another related line of work is online learning in the SSP model, which was introduced by Tarbouriech et al. [2020]. They proposed an algorithm with $\tilde{\mathcal{O}}(K^{2/3})$ regret bound. Subsequent work of Rosenberg et al. [2020] improved the regret bound to $\tilde{\mathcal{O}}(B_\star S\sqrt{AK})$. Cohen et al. [2021], Tarbouriech et al. [2021b], Chen et al. [2021] proved a minimax regret bound of $\tilde{\mathcal{O}}(B_\star\sqrt{SAK})$. However, none of these works propose a PS-type algorithm. We refer the reader to Yin et al. [2022] for offline learning of the SSP model, Rosenberg and Mansour [2020], Chen et al. [2020], Chen and Luo [2021] for the SSP model with adversarial costs and Tarbouriech et al. [2021a] for sample complexity of the SSP model with a generative model.

2 PRELIMINARIES

A Stochastic Shortest Path (SSP) model is denoted by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, c, \theta, s_{\text{init}}, g)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $c : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the cost function, $s_{\text{init}} \in \mathcal{S}$ is the initial state, $g \notin \mathcal{S}$ is the goal state, and $\theta : \mathcal{S}^+ \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ represents the transition kernel such that $\theta(s'|s, a) = \mathbb{P}(s'_t = s'|s_t = s, a_t = a)$ where $\mathcal{S}^+ = \mathcal{S} \cup \{g\}$ includes the goal state as well. Here $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$ are the state and action at time $t = 1, 2, 3, \cdots$ and $s'_t \in \mathcal{S}^+$ is the subsequent state. We assume that the initial state s_{init} is a fixed and known state and $\mathcal{S}$ and $\mathcal{A}$ are finite sets with size S and A, respectively. A stationary policy is a deterministic map $\pi : \mathcal{S} \to \mathcal{A}$ that maps a state to an action. The *value function* (also called the *cost-to-go function*) associated with policy π is a function $V^\pi(\cdot; \theta) : \mathcal{S}^+ \to [0, \infty]$ given by $V^\pi(g; \theta) = 0$ and $V^\pi(s; \theta) := \mathbb{E}[\sum_{t=1}^{\tau_\pi(s)} c(s_t, \pi(s_t))|s_1 = s]$ for $s \in \mathcal{S}$, where $\tau_\pi(s)$ is the number of steps before reaching the goal state (a random variable) if the initial state is s and policy π is followed throughout the episode. Here, we use the notation $V^\pi(\cdot; \theta)$ to explicitly show the dependence of the value function on θ. Furthermore, the optimal value function can be defined as $V(s; \theta) = \min_\pi V^\pi(s; \theta)$. Policy π is called *proper* if the goal state is reached with probability 1, starting from any initial state and following π (i.e., $\max_s \tau_\pi(s) < \infty$ almost surely), otherwise it is called *improper*.

We consider the reinforcement learning problem of an agent interacting with an SSP model $\mathcal{M} = (\mathcal{S}, \mathcal{A}, c, \theta_*, s_{\text{init}}, g)$ whose transition kernel θ_* is randomly generated according to the prior distribution μ_1 at the beginning and is then fixed. We will focus on SSP models with transition kernels in the set $\Theta_{B_\star}$ with the following standard properties:

Assumption 2.1. For all $\theta \in \Theta_{B_\star}$, the following holds: (1) there exists a proper policy, (2) for all improper policies π_θ, there exists a state $s \in \mathcal{S}$, such that $V^{\pi_\theta}(s; \theta) = \infty$, and (3) the optimal value function satisfies $\max_s V(s; \theta) \le B_\star$.

Bertsekas and Tsitsiklis [1991] prove that the first two conditions in Assumption 2.1 imply that for each $\theta \in \Theta_{B_\star}$, the optimal policy is stationary, deterministic, proper, and can be obtained by the minimizer of the *Bellman optimality equations* given by $V(s; \theta) =$

$$\min_a \Big\{ c(s, a) + \sum_{s' \in \mathcal{S}^+} \theta(s'|s, a) V(s'; \theta) \Big\}, \forall s \in \mathcal{S}. \quad (1)$$

Standard techniques such as Value Iteration and Policy Iteration can be used to compute the optimal policy if the SSP model is known [Bertsekas, 2017]. Here, we assume that $\mathcal{S}$, $\mathcal{A}$, and the cost function c are known (though the algorithm can be extended easily when unknown); and the transition kernel θ_* is unknown. Moreover, we assume that the support of the prior distribution μ_1 is a subset of $\Theta_{B_\star}$.

The agent interacts with the environment in K episodes. Each episode starts from the initial state s_{init} and ends at the goal state g (the agent may never reach the goal). At time t, the agent observes state s_t and takes action a_t. The environment then yields the next state $s'_t \sim \theta_*(\cdot|s_t, a_t)$. If the goal is reached (i.e., $s'_t = g$), then the current episode completes, a new episode starts, and $s_{t+1} = s_{\text{init}}$. If the goal is not reached (i.e., $s'_t \neq g$), then $s_{t+1} = s'_t$. The goal of the agent is to minimize the expected cumulative cost after K episodes, or equivalently, minimize the *Bayesian regret*:

$$R_K := \mathbb{E}\left[\sum_{t=1}^{T_K} c(s_t, a_t) - K V(s_{\text{init}}; \theta_*) \right],$$

where T_K is the total number of time steps before reaching the goal state for the Kth time, and $V(s_{\text{init}}; \theta_*)$ is the optimal value function from (1). Here, expectation is with respect to the prior distribution μ_1 for θ_*, the horizon T_K, the randomness in the state transitions, and the randomness of the algorithm. If the agent does not reach the goal state at any of the episodes (i.e., $T_K = \infty$), we define $R_K = \infty$.

3 THE PSRL-SSP ALGORITHM

In this section, we propose the Posterior Sampling Reinforcement Learning (PSRL-SSP) algorithm (Algorithm 1) for the SSP model. The input of the algorithm is the prior distribution μ_1. At time t, the agent maintains the posterior distribution μ_t on the unknown parameter θ_* given by $\mu_t(\Theta) = \mathbb{P}(\theta_* \in \Theta | \mathcal{F}_t)$ for any set $\Theta \subseteq \Theta_{B_\star}$. Here $\mathcal{F}_t$ is the information available at time t (i.e., the sigma algebra generated by $s_1, a_1, \cdots, s_{t-1}, a_{t-1}, s_t$). Upon observing state s'_t by taking action a_t at state s_t, the posterior can be updated according to

$$\mu_{t+1}(d\theta) = \frac{\theta(s'_t|s_t, a_t)\mu_t(d\theta)}{\int \theta'(s'_t|s_t, a_t)\mu_t(d\theta')}. \tag{2}$$

The PSRL-SSP algorithm proceeds in epochs $\ell = 1, 2, 3, \cdots$. Let t_ℓ denote the start time of epoch ℓ. In the beginning of epoch ℓ, parameter θ_ℓ is sampled from the posterior distribution μ_{t_ℓ} and the actions within that epoch are chosen according to the optimal policy with respect to θ_ℓ. Each epoch ends if either of the two stopping criteria are satisfied. The first criterion is triggered if the number of visits to the goal state during the current epoch (denoted by K_ℓ) exceeds that of the previous epoch. This ensures that $K_\ell \leq K_{\ell-1} + 1$ for all ℓ. The second criterion is triggered if the number of visits to any of the state-action pairs is doubled compared to the beginning of the epoch. This guarantees that $n_t(s, a) \leq 2n_{t_\ell}(s, a)$ for all (s, a) where $n_t(s, a) = \sum_{\tau=1}^{t-1} \mathbf{1}_{\{s_\tau = s, a_\tau = a\}}$ denotes the number of visits to state-action pair (s, a) before time t.

The second stopping criterion is similar to that used by Jaksch et al. [2010], Rosenberg et al. [2020], Agrawal and Jia [2017], and is one of the two stopping criteria in the posterior sampling algorithm (TSDE) for the infinite-horizon average-cost MDPs [Ouyang et al., 2017b]. This stopping criterion is crucial since it allows the algorithm to switch policies if the generated policy is improper and cannot reach the goal. Note that updating the policy only at the beginning of an episode (as done in the posterior sampling for finite-horizon MDPs [Osband et al., 2013]) does not work for SSP models, because if the generated policy in the beginning of the episode is improper, the goal is never reached and the regret is infinity.

The first stopping criterion is novel. A similar stopping criterion used in the posterior sampling for infinite-horizon MDPs [Ouyang et al., 2017b] is based on the length of the epochs, i.e., a new epoch starts if the length of the current epoch exceeds the length of the previous epoch. This leads to a bound of $\mathcal{O}(\sqrt{T_K})$ on the number of epochs which translates to a final regret bound of $\mathcal{O}(K^{2/3})$ in SSP models. However, our first stopping criterion allows us to bound the number of epochs by $\mathcal{O}(\sqrt{K})$ rather than $\mathcal{O}(\sqrt{T_K})$ (see Lemma 4.2). This is a key step in avoiding dependency on $c_{\min}^{-1}$ (i.e., a lower bound on the cost function) and achieve a final regret bound of $\mathcal{O}(\sqrt{K})$.

Algorithm 1 PSRL-SSP

Input: μ_1
Initialization: $t \leftarrow 1, \ell \leftarrow 0, K_{-1} \leftarrow 0, t_0 \leftarrow 0, k_{t_0} \leftarrow 0$
for episodes $k = 1, 2, \cdots, K$ **do**
  $s_t \leftarrow s_{\text{init}}$
  while $s_t \neq g$ **do**
    if $k - k_{t_\ell} > K_{\ell-1}$ *or* $n_t(s, a) > 2n_{t_\ell}(s, a)$ *for some* $(s, a) \in \mathcal{S} \times \mathcal{A}$ **then**
      $K_\ell \leftarrow k - k_{t_\ell}$
      $\ell \leftarrow \ell + 1$
      $t_\ell \leftarrow t$
      $k_{t_\ell} \leftarrow k$
      Generate $\theta_\ell \sim \mu_{t_\ell}(\cdot)$ and compute $\pi_\ell(\cdot) = \pi^*(\cdot; \theta_\ell)$ according to (1)
    Choose action $a_t = \pi_\ell(s_t)$ and observe $s'_t \sim \theta_*(\cdot|s_t, a_t)$
    Update μ_{t+1} according to (2)
    $s_{t+1} \leftarrow s'_t$
    $t \leftarrow t + 1$

Remark 3.1. PSRL-SSP only requires to know the prior distribution μ_1. Unlike Cohen et al. [2021], knowledge of $B_\star, T_\star$ (an upper bound on the expected time the optimal policy takes to reach the goal) is not needed.

Remark 3.2. Computing the posterior can be done through conjugate distributions. For a fixed state-action pair (s, a), the likelihood distribution of the next state follows a categorical distribution. Thus, the Dirichlet distribution should be chosen as the conjugate prior.

Remark 3.3. PSRL-SSP can easily deal with unknown cost functions in the exact same way as Osband et al. [2013] has done with only a constant overhead for the regret. More precisely, one can maintain a posterior distribution on both the cost function and the transition kernel separately (by choosing a normal-gamma distribution for the cost function and Dirichlet distribution for the transition kernel). Then, at the time of sampling, the algorithm samples both the transition kernel and the cost function from the posterior and computes the optimal policy for the sampled SSP. Our known cost function assumption is just for simplicity of explanation and is not a limitation of the algorithm, or its analysis.

Main Results. Our first result considers the case where the cost function is strictly positive for all state-action pairs. Subsequently, we extend the result to the general case by adding a small positive perturbation to the cost function and running the algorithm with the perturbed costs. We first assume make a standard assumption for SSP models:

Assumption 3.4. There exists $c_{\min} > 0$, such that $c(s, a) \geq c_{\min}$ for all state-action pairs (s, a).

This assumption allows us to bound the total time spent in K episodes with the total cost, i.e., $c_{\min} T_K \leq C_K$, where $C_K := \sum_{t=1}^{T_K} c(s_t, a_t)$ is the total cost during the K episodes. To facilitate the presentation of the results, we assume that $S \geq 2$, $A \geq 2$, and $K \geq S^2 A$. The first main result is as follows.

Theorem 3.5. *Suppose Assumptions 2.1 and 3.4 hold. Then, the regret of* `PSRL-SSP` *is upper bounded as*

$$R_K = \mathcal{O}\left(B_\star S\sqrt{KA}L^2 + S^2 A \sqrt{\frac{{B_\star}^3}{c_{\min}}} L^2 \right),$$

where $L = \log(B_\star SAK c_{\min}^{-1})$.

Note that when $K \gg B_\star S^2 A c_{\min}^{-1}$, the regret bound scales as $\tilde{\mathcal{O}}(B_\star S\sqrt{KA})$. A crucial point about the above result is that the dependency on $c_{\min}^{-1}$ is only in the lower order term. This allows us to extend the $\mathcal{O}(\sqrt{K})$ bound to the general case where Assumption 3.4 does not hold by using the perturbation technique of Rosenberg et al. [2020] (see Theorem 3.6). We avoid dependency on $c_{\min}^{-1}$ in the main term by use of a Bernstein-type confidence set in the analysis inspired by Rosenberg et al. [2020]. We note that using a Hoeffding-type confidence set in the analysis as in Ouyang et al. [2017b] gives a regret bound of $\mathcal{O}(\sqrt{K/c_{\min}})$ which results in $\mathcal{O}(K^{2/3})$ regret bound if Assumption 3.4 is violated.

Theorem 3.6. *Suppose Assumption 2.1 holds and let* $\tilde{L} := \log(KB_\star T_\star SA)$. *Running* `PSRL-SSP` *with costs* $c_\epsilon(s,a) := \max\{c(s,a), \epsilon\}$ *for* $\epsilon = (S^2A/K)^{2/3}$ *yields*

$$R_K = \mathcal{O}\Big(B_\star S\sqrt{KA}\tilde{L}^2 + (S^2A)^{\frac{2}{3}} K^{\frac{1}{3}} (B_\star^{\frac{3}{2}} \tilde{L}^2 + T_\star) + S^2 A T_\star^{\frac{3}{2}} \tilde{L}^2 \Big).$$

Note that when $K \gg S^2A(B_\star^3 + T_\star(T_\star/B_\star)^6)$, the regret bound scales as $\tilde{\mathcal{O}}(B_\star S\sqrt{KA})$. These results have similar regret bounds as the `Bernstein-SSP` algorithm [Rosenberg et al., 2020], and have a gap of $\sqrt{S}$ with the lower bound of $\Omega(B_\star\sqrt{SAK})$.

4 THEORETICAL ANALYSIS

In this section, we prove Theorem 3.5. Proof of Theorem 3.6 can be found in the Appendix.

A key property of posterior sampling is that conditioned on the information at time t, θ_* and θ_t have the same distribution if θ_t is sampled from the posterior distribution at time t [Osband et al., 2013]. Since `PSRL-SSP` samples θ_ℓ at the stopping time t_ℓ, we use the stopping time version of the posterior sampling property stated as follows.

Lemma 4.1 (Adapted from Lemma 2 of Ouyang et al. [2017b])**.** *Let* t_ℓ *be a stopping time with respect to the filtration* $(\mathcal{F}_t)_{t=1}^\infty$*, and* θ_ℓ *be the sample drawn from the posterior distribution at time* t_ℓ*. Then, for any measurable function* f *and any* $\mathcal{F}_{t_\ell}$*-measurable random variable* X*, we have* $\mathbb{E}[f(\theta_\ell, X)|\mathcal{F}_{t_\ell}] = \mathbb{E}[f(\theta_*, X)|\mathcal{F}_{t_\ell}]$.

We now sketch the proof of Theorem 3.5. Let $0 < \delta < 1$ be a parameter to be chosen later. We distinguish between *known* and *unknown* state-action pairs. A state-action pair (s,a) is *known* if the number of visits to (s,a) is at least $\alpha \cdot \frac{B_\star S}{c_{\min}} \log \frac{B_\star SA}{\delta c_{\min}}$ for some large enough constant α (to be determined in Lemma A.6), and *unknown* otherwise. We divide each epoch into *intervals*. The first interval starts at time $t = 1$. Each interval ends if any of the following conditions hold: (i) the total cost during the interval is at least $B_\star$; (ii) an unknown state-action pair is met; (iii) the goal state is reached; or (iv) the current epoch completes. The idea of introducing intervals is that after all state-action pairs are known, the cost accumulated during an interval is at least $B_\star$ (ignoring conditions (iii) and (iv)), which allows us to bound the number of intervals with the total cost divided by $B_\star$. Introducing intervals and distinguishing between known and unknown state-action pairs is only in the analysis and thus knowledge of $B_\star$ is not required.

Instead of bounding R_K, we bound R_M defined as

$$R_M := \mathbb{E}\left[\sum_{t=1}^{T_M} c(s_t, a_t) - KV(s_{\text{init}}; \theta_*) \right],$$

for any number of intervals M as long as K episodes are not completed. Here, T_M is the total time of the first M intervals. Let C_M denote the total cost of the algorithm after M intervals and define L_M as the number of epochs in the first M intervals. Observe that the number of times conditions (i), (ii), (iii), and (iv) trigger to start a new interval are bounded by $C_M/B_\star$, $\mathcal{O}(\frac{B_\star S^2 A}{c_{\min}} \log \frac{B_\star SA}{\delta c_{\min}})$, K, and L_M, respectively. Thus, the number of intervals is bounded as

$$M \leq \frac{C_M}{B_\star} + K + L_M + \mathcal{O}(\frac{B_\star S^2 A}{c_{\min}} \log \frac{B_\star SA}{\delta c_{\min}}). \quad (3)$$

Moreover, since the cost function is lower bounded by $c_{\min}$, we have $c_{\min} T_M \leq C_M$. Our argument proceeds as follows.[1] We bound $R_M \lesssim B_\star S\sqrt{MA}$ which implies $\mathbb{E}[C_M] \lesssim K\mathbb{E}[V(s_{\text{init}}; \theta_*)] + B_\star S\sqrt{MA}$. From the definition of intervals and once all the state-action pairs are known, the cost accumulated within each interval is at least $B_\star$ (ignoring intervals that end when the epoch or episode ends). This allows us to bound the number of intervals M with $C_M/B_\star$ (or $\mathbb{E}[C_M]/B_\star$). Solving for $\mathbb{E}[C_M]$ in the quadratic inequality $\mathbb{E}[C_M] \lesssim K\mathbb{E}[V(s_{\text{init}}; \theta_*)] + B_\star S\sqrt{MA} \lesssim K\mathbb{E}[V(s_{\text{init}}; \theta_*)] + S\sqrt{\mathbb{E}[C_M]B_\star A}$ implies that $\mathbb{E}[C_M] \lesssim K\mathbb{E}[V(s_{\text{init}}; \theta_*)] + B_\star S\sqrt{AK}$. Since this

[1] Lower order terms are neglected.

bound holds for any number of M intervals as long as K episodes are not passed, it holds for $\mathbb{E}[C_K]$ as well. Moreover, since $c_{\min} > 0$, this implies that the K episodes eventually terminate and proves the final regret bound.

Bounding the Number of Epochs. Before proceeding with bounding R_M, we first prove that the number of epochs is bounded as $\mathcal{O}(\sqrt{KSA \log T_M})$. Recall that the length of the epochs is determined by two stopping criteria. If we ignore the second criterion for a moment, the first stopping criterion ensures that the number of episodes within each epoch grows at a linear rate which implies that the number of epochs is bounded by $\mathcal{O}(\sqrt{K})$. If we ignore the first stopping criterion for a moment, the second stopping criterion triggers at most $\mathcal{O}(SA \log T_M)$ times. The following lemma shows that the number of epochs remains of the same order even if these two criteria are considered simultaneously.

Lemma 4.2. *The number of epochs is bounded as* $L_M \leq \sqrt{2SAK \log T_M} + SA \log T_M$.

We now provide the proof sketch for bounding R_M. With abuse of notation let $t_{L_M+1} := T_M + 1$ and write

$$R_M := \mathbb{E}\left[\sum_{t=1}^{T_M} c(s_t, a_t) - KV(s_{\text{init}}; \theta_*)\right]$$
$$= \mathbb{E}\left[\sum_{\ell=1}^{L_M} \sum_{t=t_\ell}^{t_{\ell+1}-1} c(s_t, a_t)\right] - K\mathbb{E}\left[V(s_{\text{init}}; \theta_*)\right].$$

Note that within epoch ℓ, action a_t is taken according to the optimal policy with respect to θ_ℓ. Thus, with the Bellman equation we can write

$$c(s_t, a_t) = V(s_t; \theta_\ell) - \sum_{s'} \theta_\ell(s'|s_t, a_t) V(s'; \theta_\ell).$$

Substituting this, and adding and subtracting $V(s_{t+1}; \theta_\ell)$ and $V(s'_t; \theta_\ell)$, decomposes R_M as

$$R_M = R_M^1 + R_M^2 + R_M^3,$$

where

$$R_M^1 := \mathbb{E}\left[\sum_{\ell=1}^{L_M} \sum_{t=t_\ell}^{t_{\ell+1}-1} \left[V(s_t; \theta_\ell) - V(s_{t+1}; \theta_\ell)\right]\right],$$

$$R_M^2 := \mathbb{E}\Bigg[\sum_{\ell=1}^{L_M} \sum_{t=t_\ell}^{t_{\ell+1}-1} \Big[V(s_{t+1}; \theta_\ell) - V(s'_t; \theta_\ell)\Big]\Bigg] - K\mathbb{E}\left[V(s_{\text{init}}; \theta_*)\right],$$

$$R_M^3 := \mathbb{E}\Bigg[\sum_{\ell=1}^{L_M} \sum_{t=t_\ell}^{t_{\ell+1}-1} \Big[V(s'_t; \theta_\ell) - \sum_{s'} \theta_\ell(s'|s_t, a_t) V(s'; \theta_\ell)\Big]\Bigg].$$

We proceed by bounding these terms separately. Proof of these lemmas can be found in the supplementary material. R_M^1 is a telescopic sum and can be bounded by the following lemma.

Lemma 4.3. *The first term* R_M^1 *is bounded as* $R_M^1 \leq B_\star \mathbb{E}[L_M]$.

To bound R_M^2, recall that $s'_t \in \mathcal{S}^+$ is the next state of the environment after applying action a_t at state s_t, and that $s'_t = s_{t+1}$ for all time steps except the last time step of an episode (right before reaching the goal). In the last time step of an episode, $s'_t = g$ while $s_{t+1} = s_{\text{init}}$. This proves that the inner sum of R_M^2 can be written as $V(s_{\text{init}}; \theta_\ell) K_\ell$, where K_ℓ is the number of visits to the goal state during epoch ℓ. Using $K_\ell \leq K_{\ell-1} + 1$ and the property of posterior sampling completes the proof. This is formally stated in the following lemma.

Lemma 4.4. *The second term* R_M^2 *is bounded as* $R_M^2 \leq B_\star \mathbb{E}[L_M]$.

The rest of the proof proceeds to bound the third term R_M^3 which contributes to the dominant term of the final regret bound. The detailed proof can be found in Lemma 4.5. Here we provide the proof sketch. R_M^3 captures the difference between $V(\cdot; \theta_\ell)$ at the next state $s'_t \sim \theta_*(\cdot|s_t, a_t)$ and its expectation with respect to the sampled θ_ℓ. Applying the Hoeffding-type concentration bounds [Weissman et al., 2003], as used by Ouyang et al. [2017b] yields a regret bound of $\mathcal{O}(K^{2/3})$ which is sub-optimal. To achieve the optimal dependency on K, we use a technique based on the Bernstein concentration bound inspired by the work of Rosenberg et al. [2020]. This requires a more careful analysis. Let $n_{t_\ell}(s, a, s')$ be the number of visits to state-action pair (s, a) followed by state s' before time t_ℓ. For a fixed state-action pair (s, a), define the Bernstein confidence set using the empirical transition probability $\widehat{\theta}_\ell(s'|s, a) := \frac{n_{t_\ell}(s,a,s')}{n_{t_\ell}(s,a)}$ as

$$B_\ell(s, a) := \Bigg\{\theta(\cdot|s, a) : |\theta(s'|s, a) - \widehat{\theta}_\ell(s'|s, a)| \leq 4\sqrt{\widehat{\theta}_\ell(s'|s, a) A_\ell(s, a)} + 28 A_\ell(s, a), \forall s' \in \mathcal{S}^+\Bigg\}. \quad (4)$$

Here $A_\ell(s, a) := \frac{\log(SAn_\ell^+(s,a)/\delta)}{n_\ell^+(s,a)}$ and $n_\ell^+(s, a) := \max\{n_{t_\ell}(s, a), 1\}$. This confidence set is similar to the one used by Rosenberg et al. [2020] and contains the true transition probability $\theta_*(\cdot|s, a)$ with high probability (see Lemma A.2). Note that $B_\ell(s, a)$ is $\mathcal{F}_{t_\ell}$-measurable which allows us to use the property of posterior sampling (Lemma 4.1) to conclude that $B_\ell(s, a)$ contains the sampled transition probability $\theta_\ell(\cdot|s, a)$ as well with high probability. With some algebraic manipulation, R_M^3 can be written as

(with abuse of notation $\ell := \ell(t)$ is the epoch at time t)

$$R_M^3 = \mathbb{E}\Bigg[\sum_{t=1}^{T_M}\sum_{s'\in\mathcal{S}^+}[\theta_*(s'|s_t,a_t)-\theta_\ell(s'|s_t,a_t)]\Bigg(V(s';\theta_\ell) - \sum_{s''\in\mathcal{S}^+}\theta_*(s''|s_t,a_t)V(s'';\theta_\ell)\Bigg)\Bigg].$$

Under the event that both $\theta_*(\cdot|s_t,a_t)$ and $\theta_\ell(\cdot|s_t,a_t)$ belong to the confidence set $B_\ell(s_t,a_t)$, Bernstein bound can be applied to obtain

$$R_M^3 \approx \mathcal{O}\left(\mathbb{E}\left[\sum_{t=1}^{T_M}\sqrt{SA_\ell(s_t,a_t)\mathbb{V}_\ell(s_t,a_t)}\right]\right) = \mathcal{O}\left(\sum_{m=1}^{M}\mathbb{E}\left[\sum_{t=t_m}^{t_{m+1}-1}\sqrt{SA_\ell(s_t,a_t)\mathbb{V}_\ell(s_t,a_t)}\right]\right),$$

where t_m denotes the start time of interval m and $\mathbb{V}_\ell$ is the empirical variance defined as

$$\mathbb{V}_\ell(s_t,a_t) := \sum_{s'\in\mathcal{S}^+}\theta_*(s'|s_t,a_t)\Bigg(V(s';\theta_\ell) - \sum_{s''\in\mathcal{S}^+}\theta_*(s''|s_t,a_t)V(s'';\theta_\ell)\Bigg)^2. \quad (5)$$

Applying Cauchy Schwarz on the inner sum twice implies that

$$R_M^3 \approx \mathcal{O}\Bigg(\sum_{m=1}^{M}\Bigg(\sqrt{S\mathbb{E}\left[\sum_{t=t_m}^{t_{m+1}-1}A_\ell(s_t,a_t)\right]}\cdot\sqrt{\mathbb{E}\left[\sum_{t=t_m}^{t_{m+1}-1}\mathbb{V}_\ell(s_t,a_t)\right]}\Bigg)\Bigg).$$

Using the fact that all the state-action pairs (s_t,a_t) within an interval except possibly the first one are known, and that the cumulative cost within an interval is at most $2B_\star$, one can bound $\mathbb{E}\left[\sum_{t=t_m}^{t_{m+1}-1}\mathbb{V}_\ell(s_t,a_t)\right] = \mathcal{O}(B_\star^2)$ (see Lemma A.5 for details). Applying Cauchy Schwarz implies $R_M^3 \approx$

$$\mathcal{O}\left(B_\star\sqrt{MS\mathbb{E}\left[\sum_{t=1}^{T_M}A_\ell(s_t,a_t)\right]}\right) \approx \mathcal{O}\left(B_\star S\sqrt{MA}\right).$$

This argument is formally presented in the following lemma.

Lemma 4.5. *The third term R_M^3 can be bounded as*

$$R_M^3 \le 288B_\star S\sqrt{MA\log^2\frac{SA\mathbb{E}[T_M]}{\delta}} + 1632B_\star S^2A\log^2\frac{SA\mathbb{E}[T_M]}{\delta} + 4SB_\star\delta\mathbb{E}[L_M].$$

Detailed proofs of all lemmas and the theorem can be found in the appendix in the supplementary material.

5 EXPERIMENTS

The literature on regret-minimization for the SSP model mostly lacks numerical evaluation except for Tarbouriech et al. [2020]. Standard OpenAI Gym environments Brockman et al. [2016] are either not designed for the SSP setting (e.g., FrozenLake-v0, CartPole-v1, and MuJoCo), or more suitable for algorithms with function approximation (e.g., Atari and Box2D). In this section, we attempt to design some benchmark environments and compare the performance of our `PSRL-SSP` algorithm with existing OFU-type algorithms in the literature. Three environments are considered: RandomMDP, GridWorld, and SSP-MountainCar.

Description of Environments. RandomMDP [Ouyang et al., 2017b, Wei et al., 2020] is an SSP with 8 states and 2 actions whose transition kernel and cost function are generated uniformly at random ($c_{\min} = 0.04$).

GridWorld [Tarbouriech et al., 2020] is a 3×4 grid (total of 12 states including the goal state) and 4 actions (LEFT, RIGHT, UP, DOWN) with $c(s,a) = 1$ for any state-action pair $(s,a) \in \mathcal{S}\times\mathcal{A}$. The agent starts from the initial state located at the top left corner of the grid, and ends in the goal state at the bottom right corner. At each time step, the agent attempts to move in one of the four directions. However, the attempt is successful only with probability 0.85. With probability 0.15, the agent takes any of the undesired directions uniformly at random. If the agent tries to move out of the boundary, the attempt will not be successful and it remains in the same position.

The SSP-MountainCar environment is a modification of the standard MountainCar-v0 environment [Moore, 1990] and simulates a car positioned between two mountains and wants to drive up the mountain on the right, however, the engine is not powerful enough to ascend directly. It needs to drive back and forth to build adequate momentum. This is a continuous state space SSP model with three actions (LEFT, RIGHT, NEUTRAL). The state is the pair of (position, velocity), where position can take values in $[-1.2, 0.6]$ and velocity can take values in $[-0.07, 0.07]$. The agent suffers a cost of 1 at each time before reaching the goal. We discretize the state space with the step size of 0.1 for the position and 0.02 for the velocity (total of $126 = 18 \times 7$ states). Note that this discretization is only from the perspective of the agent and the underlying dynamics are unchanged. Although the underlying environment is deterministic, the agent observes stochastic transitions due to the discretization. Note that the standard MountainCar-v0 environment artificially terminates the interaction between the agent and the environment after 200 steps and is much simpler than the SSP-MountainCar where the interaction only terminates if the goal is reached. Indeed, the standard RL algorithms Sutton and Barto [2018] (such as Q-learning and SARSA) that work well in the MountainCar-v0 cannot reach the goal even in the first episode in the SSP-MountainCar environment.

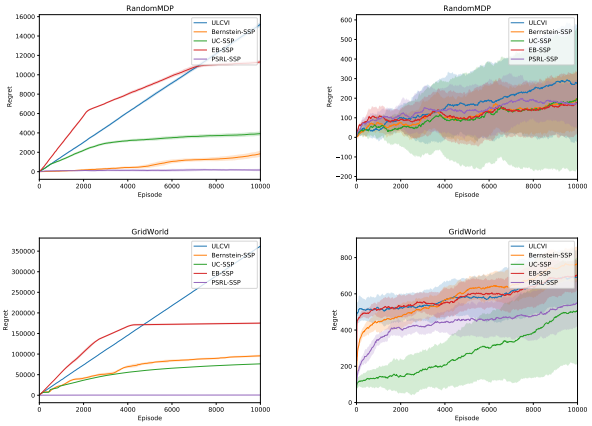

Figure 1: Cumulative regret of existing SSP algorithms on RandomMDP (top) and GridWorld (bottom) for $10,000$ episodes. The results are averaged over 10 runs and 95% confidence interval is shown with the shaded area. Our proposed PSRL−SSP algorithm outperforms all the existing algorithms considerably if the confidence intervals of other algorithms are not tuned (left plots). PSRL−SSP (with no hyper-parameter tuning) has similar performance to OFU algorithms if their confidence intervals are tuned as a hyper-parameter (right plots).

In the experiments, we evaluate the frequentist regret of PSRL−SSP for a fixed environment (i.e., the environment is not sampled from a prior distribution). A Dirichlet prior with parameters $[0.1, \cdots, 0.1]$ is considered for the transition kernel, which remain the same across environments and are not tuned as hyper-parameters. Dirichlet is a common prior in Bayesian statistics since it is a conjugate prior for categorical and multinomial distributions.

We compare the performance of our proposed PSRL−SSP against all provable existing online learning algorithms for the SSP problem (UC−SSP [Tarbouriech et al., 2020], Bernstein−SSP [Rosenberg et al., 2020], ULCVI [Cohen et al., 2021], and EB−SSP [Tarbouriech et al., 2021b]). The results are averaged over 10 independent runs. 95% confidence interval is considered to compare the performance of the algorithms. All the experiments are performed on a 2015 Macbook Pro with 2.7 GHz Dual-Core Intel Core i5 processor and 16GB RAM.

We compare PSRL−SSP with OFU algorithms in two scenarios. The first scenario, considers the case where the theoretical confidence intervals are used for the OFU algorithms (Figure 1 (left)). The second scenario is when a multiplicative coefficient (smaller than 1) is used in front of the confidence intervals for the OFU algorithms to expedite learning (Figure 1 (right)). This coefficient is tuned as a hyper-parameter. It can be seen from Figure 1 (left) that PSRL−SSP significantly outperforms all the previously proposed algorithms for the SSP problem if the theoretical confidence intervals are used. In particular, it outperforms the recently proposed ULCVI [Cohen et al., 2021] and EB−SSP [Tarbouriech et al., 2021b] which match the theoretical lower bound. Our numerical evaluation reveals that the ULCVI algorithm does not show any evidence of learning even after 80,000 episodes (not shown here). Figure 1 (right) verifies that the performance of PSRL−SSP (with no hyper parameter tuning) is similar to the tuned OFU algorithms (where confidence interval is tuned as a hyper parameter). The poor performance of OFU algorithms ensures the necessity to consider PS algorithms in practice.

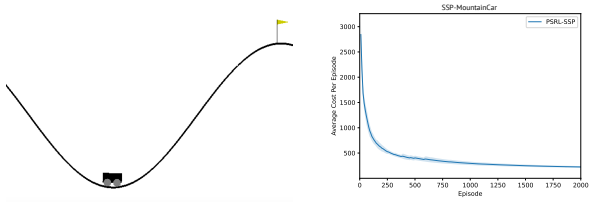

Figure 2: (left) SSP-MountainCar environment. (right) Average cost per episode of the PSRL−SSP algorithm. OFU algorithms did not learn in reasonable time (and thus not included) due to the large state space.

The gap between the performance of PSRL−SSP and OFU algorithms is even more apparent in the GridWorld environment which is more challenging compared to RandomMDP. Note that in RandomMDP, it is possible to go to the goal state from any state with only one step. This is since the transition kernel is generated uniformly at random. However, in the GridWorld environment, the agent has to take a sequence of actions to the right and down to reach the goal at the bottom right corner. Figure 1 (bottom) verifies that PSRL−SSP is able to learn this pattern significantly faster than OFU algorithms.

Figure 2 evaluates the performance of PSRL−SSP in the SSP-MountainCar environment which has a much larger state space. The large state space of this environment prevents OFU algorithms from learning in reasonable amount of time (and thus not shown in the figure). However, PSRL−SSP improves quickly after a few episodes.

These results confirm the intuition that OFU-type algorithms are too conservative in uncertainty estimation, whereas PS-type algorithms are statistically more efficient and hence perform better empirically across almost all settings.

CONCLUSIONS

In this paper, we have proposed the first posterior sampling-based reinforcement learning algorithm for the SSP models with unknown transition probabilities. The algorithm is very simple as compared to the optimism-based algorithm proposed for SSP models recently [Tarbouriech et al.,

2020, Rosenberg et al., 2020, Cohen et al., 2021, Tarbouriech et al., 2021b]. It achieves a Bayesian regret bound of $\tilde{\mathcal{O}}(B_\star S\sqrt{AK})$, where $B_\star$ is an upper bound on the expected cost of the optimal policy, S is the size of the state space, A is the size of the action space, and K is the number of episodes. This has a $\sqrt{S}$ gap from the best known bound for an optimism-based algorithm but numerical experiments suggest a better performance in practice. A next step would be to extend the algorithm to continuous state and action spaces, and to propose model-free algorithms for such settings. Designing posterior sampling-based model-free algorithms for even average MDPs remains an open problem. Another interesting future direction is to extend ideas from Tiapkin et al. [2022] to obtain frequentist regret bound for posterior-sampling based algorithms in the SSP setting.

Acknowledgements HL is supported by NSF Award IIS-1943607 and a Google Research Scholar Award. RJ is supported by NSF ECCS-2025732 and ONR N00014-20-1-2258 awards.

References

Marc Abeille and Alessandro Lazaric. Thompson sampling for linear-quadratic control problems. In *Artificial Intelligence and Statistics*, pages 1246–1254. PMLR, 2017.

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1. JMLR Workshop and Conference Proceedings, 2012.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pages 127–135. PMLR, 2013.

Shipra Agrawal and Randy Jia. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In *Advances in Neural Information Processing Systems*, pages 1184–1194, 2017.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 263–272. JMLR. org, 2017.

Dragan Banjević and Michael Jong Kim. Thompson sampling for stochastic control: The continuous parameter case. *IEEE Transactions on Automatic Control*, 64(10): 4137–4152, 2019.

Dimitri P Bertsekas. Dynamic programming and optimal control, vol i and ii, 4th edition. *Belmont, MA: Athena Scientific*, 2017.

Dimitri P Bertsekas and John N Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, 24:2249–2257, 2011.

Liyu Chen and Haipeng Luo. Finding the stochastic shortest path with low regret: The adversarial cost and unknown transition case. *arXiv preprint arXiv:2102.05284*, 2021.

Liyu Chen, Haipeng Luo, and Chen-Yu Wei. Minimax regret for stochastic shortest path with adversarial costs and known transition. *arXiv preprint arXiv:2012.04053*, 2020.

Liyu Chen, Mehdi Jafarnia-Jahromi, Rahul Jain, and Haipeng Luo. Implicit finite-horizon approximation and efficient optimal algorithms for stochastic shortest path. *Advances in Neural Information Processing Systems*, 34, 2021.

Alon Cohen, Yonathan Efroni, Yishay Mansour, and Aviv Rosenberg. Minimax regret for stochastic shortest path. *arXiv preprint arXiv:2103.13056*, 2021.

Raphaël Fonteneau, Nathan Korda, and Rémi Munos. An optimistic posterior sampling strategy for bayesian reinforcement learning. In *NIPS 2013 Workshop on Bayesian Optimization (BayesOpt2013)*, 2013.

Ronan Fruit, Matteo Pirotta, Alessandro Lazaric, and Ronald Ortner. Efficient bias-span-constrained exploration-exploitation in reinforcement learning. In *International Conference on Machine Learning*, pages 1573–1581, 2018.

Aditya Gopalan and Shie Mannor. Thompson sampling for learning parameterized markov decision processes. In *Conference on Learning Theory*, pages 861–898. PMLR, 2015.

Mehdi Jafarnia-Jahromi, Rahul Jain, and Ashutosh Nayyar. Learning zero-sum stochastic games with posterior sampling. *arXiv preprint arXiv:2109.03396*, 2021.

Mehdi Jafarnia-Jahromi, Rahul Jain, and Ashutosh Nayyar. Online learning for unknown partially observable mdps. *International Conference on Artificial Intelligence and Statistics*, 2022.

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

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4863–4873, 2018.

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *International conference on algorithmic learning theory*, pages 199–213. Springer, 2012.

Michael Jong Kim. Thompson sampling for stochastic control: The finite parameter case. *IEEE Transactions on Automatic Control*, 62(12):6415–6422, 2017.

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

Andrew William Moore. Efficient memory-based learning for robot control. 1990.

Ian Osband and Benjamin Van Roy. Why is posterior sampling better than optimism for reinforcement learning? In *International Conference on Machine Learning*, pages 2701–2710. PMLR, 2017.

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

Yi Ouyang, Mukul Gagrani, and Rahul Jain. Learning-based control of unknown linear systems with thompson sampling. *arXiv preprint arXiv:1709.04047*, 2017a.

Yi Ouyang, Mukul Gagrani, Ashutosh Nayyar, and Rahul Jain. Learning unknown markov decision processes: A thompson sampling approach. In *Advances in Neural Information Processing Systems*, pages 1333–1342, 2017b.

Aviv Rosenberg and Yishay Mansour. Stochastic shortest path with adversarially changing costs. *arXiv preprint arXiv:2006.11561*, 2020.

Aviv Rosenberg, Alon Cohen, Yishay Mansour, and Haim Kaplan. Near-optimal regret bounds for stochastic shortest path. In *International Conference on Machine Learning*, pages 8210–8219. PMLR, 2020.

Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on thompson sampling. *arXiv preprint arXiv:1707.02038*, 2017.

Steven L Scott. A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.

Malcolm Strens. A bayesian framework for reinforcement learning. In *ICML*, volume 2000, pages 943–950, 2000.

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

Jean Tarbouriech, Evrard Garcelon, Michal Valko, Matteo Pirotta, and Alessandro Lazaric. No-regret exploration in goal-oriented reinforcement learning. In *International Conference on Machine Learning*, pages 9428–9437. PMLR, 2020.

Jean Tarbouriech, Matteo Pirotta, Michal Valko, and Alessandro Lazaric. Sample complexity bounds for stochastic shortest path with a generative model. In *Algorithmic Learning Theory*, pages 1157–1178. PMLR, 2021a.

Jean Tarbouriech, Runlong Zhou, Simon S Du, Matteo Pirotta, Michal Valko, and Alessandro Lazaric. Stochastic shortest path: Minimax, parameter-free and towards horizon-free regret. *arXiv preprint arXiv:2104.11186*, 2021b.

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

Daniil Tiapkin, Denis Belomestny, Daniele Calandriello, Éric Moulines, Remi Munos, Alexey Naumov, Mark Rowland, Michal Valko, and Pierre Ménard. Optimistic posterior sampling for reinforcement learning with few samples and tight guarantees. *arXiv preprint arXiv:2209.14414*, 2022.

Chen-Yu Wei, Mehdi Jafarnia-Jahromi, Haipeng Luo, Hiteshi Sharma, and Rahul Jain. Model-free reinforcement learning in infinite-horizon average-reward markov decision processes. In *International Conference on Machine Learning*, pages 10170–10180. PMLR, 2020.

Chen-Yu Wei, Mehdi Jafarnia-Jahromi, Haipeng Luo, and Rahul Jain. Learning infinite-horizon average-reward mdps with linear function approximation. *International Conference on Artificial Intelligence and Statistics*, 2021.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the l1 deviation of the empirical distribution. *Hewlett-Packard Labs, Tech. Rep*, 2003.

Ming Yin, Wenjing Chen, Mengdi Wang, and Yu-Xiang Wang. Offline stochastic shortest path: Learning, evaluation and towards optimality. In *Uncertainty in Artificial Intelligence*, pages 2278–2288. PMLR, 2022.