Reinforcement Learning for Markovian Bandits: Is Posterior Sampling more Scalable than Optimism?

Nicolas Gast, Bruno Gaujal, Kimang Khun

Reinforcement Learning for Markovian Bandits: Is Posterior Sampling more Scalable than Optimism?

Nicolas Gast, Bruno Gaujal, Kimang Khun
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, 38000 Grenoble, France
{nicolas.gast,bruno.gaujal,kimang.khun}@inria.fr

Abstract

We study learning algorithms for the classical Markovian bandit problem with discount. We explain how to adapt PSRL [24] and UCRL2 [2] to exploit the problem structure. These variants are called MB-PSRL and MB-UCRL2. While the regret bound and runtime of vanilla implementations of PSRL and UCRL2 are exponential in the number of bandits, we show that the episodic regret of MB-PSRL and MB-UCRL2 is $\tilde{O}(S\sqrt{nK})$ where K is the number of episodes, n is the number of bandits and S is the number of states of each bandit (the exact bound in S, n and K is given in the paper). Up to a factor $\sqrt{S}$, this matches the lower bound of $\Omega(\sqrt{SnK})$ that we also derive in the paper. MB-PSRL is also computationally efficient: its runtime is linear in the number of bandits. We further show that this linear runtime cannot be achieved by adapting classical non-Bayesian algorithms such as UCRL2 or UCBVI to Markovian bandit problems. Finally, we perform numerical experiments that confirm that MB-PSRL outperforms other existing algorithms in practice, both in terms of regret and of computation time.

1 Introduction

Markov Decision Processes (MDPs) are a powerful model to solve stochastic optimization problems. They suffer, however, from what is called the *curse of dimensionality*, which basically says that the state size of the Markov process is exponential in the number of the system components so that the complexity of computing an optimal policy and its value are exponential. As for existing general reinforcement learning algorithms, they all have a regret and a runtime exponential in the number of components, so they also suffer from the same curse.

Very few MDPs are known to escape from this curse of dimensionality. The most famous example is certainly the Markovian bandit problem for which an optimal policy and its value can be computed in $O(n)$, where n is the number of bandits: The optimal policy can be computed by using the Gittins indices (computed locally) and its value can be computed by using retirement values (see for example [36]).

In this paper, we study the properties of two learning algorithms: MB-PSRL and MB-UCRL2 (where MB stands for Markovian Bandit). MB-PSRL is PSRL [24] applied to Markovian bandit problem and MB-UCRL2 is a modification to UCRL2 [2]. We propose a regret definition for discounted MDP and show that both algorithms have sub-linear regret in the number of episodes and bandits. Then, we analyze their computational complexity and provide an example showing that being optimistic in each bandit is not optimistic in the global MDP. We argue that that any learning algorithm based on optimism is likely to have a runtime exponential in the number of bandits unless an oracle is given. This shows the superiority of posterior sampling approach over the optimism because the runtime of MB-PSRL is just linear in the number of bandits, so that it escapes from the curse of dimensionality.

Preprint. Under review.

We conduct a series of numerical experiments to compare the performance of MB-PSRL and MB-UCRL2 with vanilla implementations of PSRL and UCRL2, and a learning algorithm specific to Markovian bandit problems (GittinsQlearning, that is introduced in [8]). They confirm the good behavior of MB-PSRL, both in terms of regret and computational complexity.

Related Work We focus on Markovian bandit problem with discount factor $\beta < 1$ and all reward functions and transition matrices $(\boldsymbol{r}_a, Q_a)_{a\in\{1,\dots,n\}}$ are unknown. A generic approach is to ignore the bandit structure and view the Markovian bandit problem as a generic MDP and use reinforcement learning algorithms to learn an optimal policy. There are two types of such algorithms. The first type uses the *optimism in face of uncertainty* (OFU) principle. OFU methods build a confidence set for the unknown MDP and compute an optimal policy of the "best" MDP in the confidence set, *e.g.*, [2, 5, 4, 11]. A popular choice is UCRL2 [2]. The second type uses a Bayesian approach, the posterior sampling method introduced in [30], like PSRL. Such algorithms keep a posterior distribution over possible MDPs and execute the optimal policy of a sampled MDP, see *e.g.*, [29, 24, 15, 25]. All these algorithms, based on OFU or on Bayesian principles, have sub-linear bounds on the regret, which means that they provably learn the optimal policy. Yet, these bounds grow exponentially with the number of bandits.

Our work is not the first attempt to exploit the structure of an MDP to improve learning. Factored MDPs (the state space can be factored into n components) are investigated in [16], where asymptotic convergence to the optimal policy is proved to scale polynomially in the number of components. The regret of learning algorithms in factored MDP with a factored action space is considered in [22, 37, 27, 31]. Our work differs substantially from these. First, the Markovian bandit problem is not a factored MDP because the action space is global and cannot be factored. Second, our reward is discounted over an infinite horizon while factored MDPs have been analyzed with no discount. Finally, the factored MDP framework assumes that the successive optimal policies are computed by an unspecified solver. There is no guarantee that the time complexity of this solver scales linearly with the number of components, especially for OFU-based algorithms. For Markovian bandits, we get an additional leverage: the Gittins index policy is known to be an optimal policy and its computational complexity is linear in the number of bandits. This is exploited in this paper to design a posterior sampling algorithm with a linear time complexity.

Since index policies scale with the number of bandits, using Q-learning approaches to learn such a policy is also popular *e.g.*, [3, 12, 8]. The authors of [8] address the same Markovian bandit problem as we do: their algorithm learns the optimal value in the restart-in-state MDP [19] for each bandit and uses Softmax exploration to solve the exploration-exploitation dilemma. As mentioned on page 250 in [1], however, there exist no finite-time regret bounds for this algorithm, and we argue in Section 6 that its regret is likely to grow linearly. Furthermore, tuning its hyperparameters (learning rate and temperature) is rather delicate and unstable in practice.

2 Markovian Bandits

In this section, we introduce the Markovian bandit problem and recall the notion of Gittins index policy and its optimality when the parameters $(\boldsymbol{r}_a, Q_a)$ of all bandits are known.

2.1 Definitions and Main Notations

We consider a Markovian bandit problem with n bandits. Each bandit $\langle \mathcal{S}_a, \boldsymbol{r}_a, Q_a\rangle$ for $a \in \{1,\dots,n\} =: [n]$ is a Markov reward process with a finite state space $\mathcal{S}_a$ of size S. Each bandit has a mean reward vector, $\boldsymbol{r}_a \in [0,1]^S$, and a transition matrix Q_a. When Bandit a is activated in state $x_a \in \mathcal{S}_a$, it moves to state $y_a \in \mathcal{S}_a$ with probability $Q_a(x_a, y_a)$. This provides a reward whose expected value is $r_a(x_a)$. Without loss of generality, we assume that the state spaces of the bandits are pairwise distinct: $\mathcal{S}_a \cap \mathcal{S}_b = \emptyset$ for $a \neq b$. In the following, the state of a bandit a will always be denoted with an index a: we will denote such a state by x_a or y_a. As state spaces are disjoint, this allows us to simplify the notation by dropping the index a from the reward and transition matrix: when convenient, we will denote them by $r(x_a)$ instead of $r_a(x_a)$ and by $Q(x_a, y_a)$ instead of $Q_a(x_a, y_a)$ since no confusion is possible.

At time 0, the global state $\boldsymbol{X}_0$ is distributed according to some initial distribution ρ over the global state space $\mathcal{E} = \mathcal{S}_1 \times \ldots \times \mathcal{S}_n$. At time t, the decision maker observes the states[1] of all bandits, $\boldsymbol{X}_t = (X_{t,1} \ldots X_{t,n})$, and chooses which bandit A_t to activate. This problem can be cast as a MDP – that we denote by $\mathcal{M}$ – with state space $\mathcal{E}$ and action space $[n]$. Let $a \in [n]$ and $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{E}$. If the state at time t is $\boldsymbol{X}_t = \boldsymbol{x}$, the chosen bandit is $A_t = a$, then the agent receives a random reward R_t drawn from some distribution on $[0, 1]$ with mean $r(x_a)$ and the MDP $\mathcal{M}$ transitions to state $\boldsymbol{X}_{t+1} = \boldsymbol{y}$ with probability $P(\boldsymbol{x}, a, \boldsymbol{y})$ that satisfies:

$$P(\boldsymbol{x}, a, \boldsymbol{y}) = \begin{cases} Q(x_a, y_a) & \text{if } x_b = y_b \text{ for all } b \neq a; \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

That is, the active bandit makes a transition while the other bandits remain in the same state.

Let Π be the set of deterministic policies, *i.e.,* the set of functions $\pi : \mathcal{E} \mapsto [n]$ and let H be a time-horizon geometrically distributed with parameter $1 - \beta > 0$. For the MDP $\mathcal{M}$, we denote by $V^\pi_{\mathcal{M}}(\boldsymbol{x})$ the expected cumulative reward of $\mathcal{M}$ under policy π starting from an initial state $\boldsymbol{x}$:

$$V^\pi_{\mathcal{M}}(\boldsymbol{x}) = \mathbb{E}\left[\sum_{t=0}^{H-1} R_t \mid \boldsymbol{X}_0 = \boldsymbol{x}, A_t = \pi(\boldsymbol{X}_t)\right]. \tag{2}$$

Note that by definition of H, $V^\pi_{\mathcal{M}}(\boldsymbol{x})$ is equal to the discounted reward over an infinite horizon: $V^\pi_{\mathcal{M}}(\boldsymbol{x}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t R_t \mid \boldsymbol{X}_0 = \boldsymbol{x}, A_t = \pi(\boldsymbol{X}_t)\right]$. By a small abuse of notation, we also denote by $V^\pi_{\mathcal{M}}(\rho)$ the expected reward when the initial state is randomly generated according to ρ: $V^\pi_{\mathcal{M}}(\rho) = \mathbb{E}\left[V^\pi_{\mathcal{M}}(\boldsymbol{X}_0) \mid \boldsymbol{X}_0 \sim \rho\right]$. A policy π_* is optimal if $V^{\pi_*}_{\mathcal{M}}(\boldsymbol{x}) \geq V^\pi_{\mathcal{M}}(\boldsymbol{x})$ for all $\pi \in \Pi$ and $\boldsymbol{x} \in \mathcal{E}$. By [26], such a policy exists and does not depend on $\boldsymbol{x}$ (or ρ).

2.2 Gittins Index Policy

When $(\boldsymbol{r}_a, Q_a)_{a \in [n]}$ are known, it is possible to compute an optimal policy using the so called Gittins indices: Gittins defines in [14] the *Gittins index* for any bandit a in state $x_a \in \mathcal{S}_a$ as

$$\mathrm{GIndex}(x_a) = \sup_{\tau > 0} \frac{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \beta^t r_a(Z_t) \mid Z_0 = x_a\right]}{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \beta^t \mid Z_0 = x_a\right]}, \tag{3}$$

where Z is a Markov chain whose transitions are given by Q_a and τ can be any stopping time adapted to the natural filtration of $(Z_t)_{t \geq 0}$. So, Gittins index can be considered as the maximal reward density over time of a bandit at the given state.

It is shown in [14] that always activating the bandit having the largest current index is an optimal policy. Such a policy can be computed very efficiently: The computation of the indices of a bandit with S states can be done in $O(S^3)$ arithmetic operations, which means that the computation of the Gittins index policy is linear in the number of bandits as it takes $O(nS^3)$ arithmetic operations. For more details about Gittins indices and optimality, we refer to [34, 13]. For a survey on how to compute Gittins indices, we refer to [7].

3 Online Learning and Episodic Regret

We now consider that the decision maker does not know the transition matrices nor the rewards. Our goal is to design a reinforcement learning algorithm that learns the optimal policy from past observations. Similarly to what is done for finite-horizon reinforcement learning with deterministic horizon – see *e.g.*, [24, 4, 38, 18] – we consider a decision maker that faces a sequence of independent replicas of the same Markovian bandit problem, where the transitions and the rewards are drawn independently for each episode. What is new here is that the time horizon H is random and has a geometric distribution. It is drawn independently for each episode. This implies that Gittins index policy is optimal for a decision maker that would know the transition matrices and rewards.

In what follows, we consider *episodic learning algorithms*. Let $H_1, \ldots, H_k$ be the sequence of random episode lengths and let $t_k := \sum_{i=1}^{k-1} H_i$ be the starting time of the kth episode. Let

[1] Throughout the paper, we use capital letters (like X_t) to denote random variables and small letter (like $\boldsymbol{x}$) to denote their realizations. Bold letters ($\boldsymbol{X}_t$ or $\boldsymbol{x}$) design vectors. Normal letters ($X_{t,a}$ or x_a) are for scalar values.

$\mathcal{O}_{k-1} := (\boldsymbol{X}_0, A_0, R_0, \ldots, \boldsymbol{X}_{t_k-1}, A_{t_k-1}, R_{t_k-1})$ denote the observations made prior and up to episode k. An *Episodic Learning Algorithm* $\mathcal{L}$ is a function that maps observations $\mathcal{O}_{k-1}$ to $\mathcal{L}(\mathcal{O}_{k-1})$, a probability distribution over all policies. At the beginning of episode k, the algorithm samples $\pi_k \sim \mathcal{L}(\mathcal{O}_{k-1})$ and uses this policy during the whole kth episode. Note that one could also design algorithms where learning takes place inside each episode. We will see later that episodic learning as described here is enough to design algorithms that are essentially optimal, in the sense given by Theorems 1, 2 and 3.

For an instance $\mathcal{M}$ of a Markovian bandit problem and a total number of episodes K, we denote by $\mathrm{Reg}(K, \mathcal{L}, \mathcal{M})$ the expected regret of a learning algorithm $\mathcal{L}$, defined as

$$\mathrm{Reg}(K, \mathcal{L}, \mathcal{M}) := \sum_{k=1}^{K} \Delta_k, \text{where } \Delta_k = \sup_{\pi \in \Pi} \mathbb{E}[V_{\mathcal{M}}^{\pi}(\rho)] - \mathbb{E}[V_{\mathcal{M}}^{\pi_k}(\rho) \mid \pi_k \sim \mathcal{L}(\mathcal{O}_{k-1})]. \tag{4}$$

A no-regret algorithm is an algorithm $\mathcal{L}$ such that its regret $\mathrm{Reg}(K, \mathcal{L}, \mathcal{M})$ grows sub-linearly in the number of episodes K. This implies that $\Delta_k := \mathbb{E}\left[(V_{\mathcal{M}}^{\pi_*}(\rho) - V_{\mathcal{M}}^{\pi_k}(\rho))\right]$ – the expected regret over episode k – converges to 0 as k goes to infinity. By the definition of $V_{\mathcal{M}}^{\pi}$ in (2) where the horizon $H \sim Geo(1-\beta)$, such an algorithm learns an optimal policy in the MDP with discount factor β.

4 Learning Algorithms for Markovian Bandits

In what follows, we present two algorithms having a regret that grows like $\tilde{O}(\sqrt{nK})$. Both algorithms are tailored to Markovian bandit problems to overcome the exponentiality in regret but one of them uses an extended value iteration with a high computational complexity (more in this in Section 6).

MB-PSRL (Algorithm 1) MB-PSRL is a version of PSRL that uses a prior that is tailored for Markovian bandits and behaves according to Gittins index policy. MB-PSRL starts with a prior distribution ϕ_a over the parameters, $(\boldsymbol{r}_a, Q_a)$. At the start of each episode k, MB-PSRL computes a posterior distribution of parameters $\phi_a(\cdot \mid \mathcal{O}_{k-1})$ for each bandit $a \in [n]$ and samples parameters $(\boldsymbol{r}_{k,a}, Q_{k,a})$ from $\phi_a(\cdot \mid \mathcal{O}_{k-1})$. Then, MB-PSRL uses $(\boldsymbol{r}_{k,a}, Q_{k,a})_{a\in[n]}$ to compute the Gittins index policy π_k that is optimal for the sampled problem. The policy π_k is then used for the whole episode k. The only hyperparameter of MB-PSRL is the prior distribution ϕ. As we see in Appendix F, MB-PSRL seems robust to the choice of the prior distribution, even if a coherent prior gives a better performance than a misspecified prior, similarly to what happens for Thompson's sampling [28].

MB-UCRL2 (Algorithm 2) MB-UCRL2 is a modification of UCRL2 to take into account the structure of Markovian bandit problems. For a given state $x_a \in \mathcal{S}_a$, let $\hat{r}_k(x_a)$, $\hat{Q}_k(x_a, \cdot)$ be the empirical means of $r(x_a)$ and $Q(x_a, \cdot)$ and let $N_k(x_a)$ be the number of times that Bandit a is activated before episode k while being in state x_a. We define the confidence bonuses $\epsilon_k^r(x_a) = \sqrt{\frac{7\log(2Snt_k/\delta)}{2\max\{1, N_k(x_a)\}}}$ and $\epsilon_k^Q(x_a) = \sqrt{\frac{14S\log(2nt_k/\delta)}{\max\{1, N_k(x_a)\}}}$. This defines a confidence set $\mathbb{M}$ such that a Markovian bandit problem $\bar{\mathcal{M}}$ is in $\mathbb{M}$ if for all $a \in [n]$ and $x_a \in \mathcal{S}_a$:

$$|r(x_a) - \hat{r}(x_a)| \leq \epsilon^r(x_a) \text{ and } \|Q(x_a, \cdot) - \hat{Q}(x_a, \cdot)\|_1 \leq \epsilon^Q(x_a). \tag{5}$$

MB-UCRL2 then chooses a policy π_k that is optimal for the most optimistic problem $\bar{\mathcal{M}}_k \in \mathbb{M}$:

$$\pi_k \in \arg\max_{\pi} \sup_{\bar{\mathcal{M}} \in \mathbb{M}} V_{\bar{\mathcal{M}}}^{\pi}(\rho). \tag{6}$$

Compared to a vanilla implementation of UCRL2, MB-UCRL2 is specifically tailored to Markovian bandits because it uses the structure of the Markovian bandit problem: The constraints (5) are on Q whereas vanilla UCRL2 uses constraints on the full matrix P (defined in (1)). This leads MB-UCRL2 to use the bonus term that scales as $\sqrt{S/N_k(x_a)}$ whereas vanilla UCRL2 would use the term in $\sqrt{S^n/N_k(\boldsymbol{x}, a)}$. Vanilla UCRL2 is described in more details in Appendix E.

5 Regret Analysis

In this section, we first present upper bounds on the Bayesian regret of MB-PSRL and the expected regret of MB-UCRL2. These bounds are sub-linear in the number of episodes (hence both algorithms are no-regret algorithms) and sub-linear in the number of bandits. We then derive a minimax lower bound on the regret of any learning algorithm for the Markovian bandit problem.

Algorithm 1 MB-PSRL

Data: Prior distribution ϕ_a for $a \in [n]$, discount factor β, initial distribution ρ, $t_1 = 0$.

for episodes $k = 1, 2, \ldots$ **do**
  Sample $(\boldsymbol{r}_{k,a}, Q_{k,a}) \sim \phi_a(\cdot|\mathcal{O}_{k-1})$ for each $a \in [n]$.
  Compute Gittins policy π_k for the sampled problem.
  Set $t_k \leftarrow \sum_{i=1}^{k-1} H_i$. Sample an initial state $\boldsymbol{X}_{t_k} \sim \rho$ and an episode length $H_k \sim \mathrm{Geom}(1-\beta)$.
  for $t \leftarrow t_k$ **to** $t_k + H_k - 1$ **do**
    Activate bandit $A_t = \pi_k(\boldsymbol{X}_t)$.
    Observe R_t and $\boldsymbol{X}_{t+1}$.
  end for
end for

Algorithm 2 MB-UCRL2

Data: Confidence level $1-\delta$, discount factor β, initial distribution ρ, $t_1 = 0$.
for episodes $k = 1, 2, \ldots$ **do**
  Compute plausible MDP set $\mathbb{M}_k$ based on $\mathcal{O}_{k-1}$.
  Compute optimistic MDP $\bar{\mathcal{M}}_k \in \mathbb{M}_k$ and π_k by Extended Value Iteration [2].
  Set $t_k \leftarrow \sum_{i=1}^{k-1} H_i$. Sample an initial state $\boldsymbol{X}_{t_k} \sim \rho$ and an episode length $H_k \sim \mathrm{Geom}(1-\beta)$.
  for $t \leftarrow t_k$ **to** $t_k + H_k - 1$ **do**
    Activate bandit $A_t = \pi_k(\boldsymbol{X}_t)$.
    Observe R_t and $\boldsymbol{X}_{t+1}$.
  end for
end for

5.1 Upper Bounds on Regret

Suppose that the unknown MDP $\mathcal{M}$ is drawn from a prior distribution ϕ. The *Bayesian regret* of a learning algorithm $\mathcal{L}$ is $\mathrm{BayReg}(K, \mathcal{L}, \phi) = \mathbb{E}[\mathrm{Reg}(K, \mathcal{L}, \mathcal{M})]$, where the expectation is taken over all possible values of $\mathcal{M} \sim \phi$.

Theorem 1 (Regret of MB-PSRL)**.** *If the Markovian bandit model $\mathcal{M}$ is generated according to the prior distribution ϕ used by MB-PSRL then there exists a universal constant C independent of the model (i.e., it does not depend on S, n, K and β) such that, for all $\epsilon > 0$, the Bayesian regret of MB-PSRL is bounded by*

$$\mathrm{BayReg}(K, \textit{MB-PSRL}, \phi) \leq C(\sqrt{S} + \sqrt{\log(SnK)} + \frac{1}{\epsilon}) \times f(S, n, K, \beta, \epsilon),$$

where $f(S, n, K, \beta, \epsilon) = \left(Sn \left(\frac{\log K}{1-\beta}\right)^{\epsilon+2} + \sqrt{SnK} \left(\frac{\log K}{1-\beta}\right)^{\epsilon+3/2} \right)$.

Theorem 2 (Regret of MB-UCRL2)**.** *There exists a universal constant C' such that for any Markovian bandit problem $\mathcal{M}$ and $\epsilon > 0$, the regret of MB-UCRL2 is bounded with probability at least $1 - \delta$:*

$$\mathrm{Reg}(K, \textit{MB-UCRL2}, \mathcal{M}) \leq C'(\sqrt{S} + \sqrt{\log(SnK/\delta)} + \frac{1}{\epsilon}) \times f(S, n, K, \beta, \epsilon),$$

where $f(S, n, K, \beta, \epsilon)$ is given in Theorem1.

Sketch of proof for Theorem 1. The full proof is given in Appendix A. We explain here the main steps. Let π_* be the optimal policy of the true MDP $\mathcal{M}$ and π_k the optimal policy for the sampled MDP $\mathcal{M}_k$ at episode k. By definition, the expected regret over episode k, Δ_k, equals

$$\underbrace{\mathbb{E}\left[V_{\mathcal{M}}^{\pi_*}(\boldsymbol{X}_{t_k}) - V_{\mathcal{M}_k}^{\pi_k}(\boldsymbol{X}_{t_k})\right]}_{(A)} + \underbrace{\mathbb{E}\left[V_{\mathcal{M}_k}^{\pi_k}(\boldsymbol{X}_{t_k}) - V_{\mathcal{M}}^{\pi_k}(\boldsymbol{X}_{t_k})\right]}_{(B)}. \tag{7}$$

As our setting is Bayesian, the first term (A) is zero (see Lemma 1). We then prove that (B) is bounded by the distance between the sampled MDP $\mathcal{M}_k$ and the true MDP $\mathcal{M}$. This distance is bounded by using a concentration argument (Lemma 3) based on Hoeffding's and Weissman's inequalities. Hoeffding's inequality is valid if the realization of the rewards are in $[0, 1]$. This restriction can be lifted by considering sub-Gaussian random rewards. The only difference in the proof would be to replace Hoeffding's inequality by a similar inequality that is valid for sub-Gaussian random variables, see *e.g.*, [33].

The main technical hurdle then is to deal with the K random episodes $H_1, \ldots, H_k$. This is new in our approach compared to the classical analysis of finite horizons regrets. To bound this, one

needs to bound terms of the form $\mathbb{E}\left[\max_{1\leq k\leq K}(H_k)^\alpha \log H_k\right]$ with $\alpha \in \{1.5, 2\}$. To do so, we first use that $\log(x) \leq x^\epsilon/(e\epsilon)$ for any $\epsilon > 0$ (Lemma 4) to remove the logarithmic term and then use the geometric distribution of H_k to show that $\mathbb{E}\left[\max_{1\leq k\leq K}(H_k)^{\alpha+\epsilon}\right] = O((\frac{\log K}{1-\beta})^{\alpha+\epsilon})$ (see Lemma 5). □

Sketch of proof for Theorem 2. As argued in Theorem 1 of [23], the main difference between the proofs of PSRL and UCRL2 is to deal with the term (A) of (7). For MB-UCRL2, the definition of the optimistic policy π_k says that if $\mathcal{M} \in \mathbb{M}$, then

$$\sup_{\bar{\mathcal{M}}\in\mathbb{M}} V^{\pi_k}_{\bar{\mathcal{M}}}(\rho) \geq \sup_{\pi} V^{\pi}_{\mathcal{M}}(\rho), \tag{8}$$

so that the term (A) of (7) is non-positive as soon as $\mathcal{M} \in \mathbb{M}$. By definition of ϵ^r and ϵ^Q, this occurs with high probability. Appendix C gives a detailed proof. □

These theorems call for several comments.
- First, it shows that when $K \geq Sn/(1-\beta)$, the regret of both algorithms is smaller than

$$\tilde{O}\left(\frac{S\sqrt{nK}}{(1-\beta)^{\epsilon+3/2}}\right), \tag{9}$$

where the notation $\tilde{O}$ means that all logarithmic terms are removed. Hence, the regret of both algorithms is sub-linear in the number of episodes K which means that both algorithms are no-regret algorithms. This regret bound is sub-linear in the number of bandits which is very significant in practice when facing a large number of bandits. Note that directly applying PSRL or UCRL2 would lead to a regret in $\tilde{O}\left(S^n\sqrt{nK}\right)$, which is exponential in n.

- Second, the upper bound on the expected regret of MB-UCRL2 is a guarantee for a specific problem $\mathcal{M}$ while the bound on Bayesian regret of MB-PSRL is a guarantee in average overall the problems drawn from the prior ϕ. Hence, the bound of MB-UCRL2 is a stronger guarantee compared to the one of MB-PSRL. At this point MB-UCRL2 seems to have the upper hand over MB-PSRL. However, these bounds do not tell the whole story and the numerical experiments reported in Section 7 show that MB-PSRL has a smaller regret in practice, even when the problem does not follow the right prior. We will also see in Section 6 that MB-UCRL2 and any other optimistic algorithm applied to Markovian bandit problem have computational complexity that is exponential in the number of bandits while MB-PSRL has a linear complexity.

- Finally, our bound (9) is linear in S, the state size of each bandit, because our proof follows the approach used in [24]. Using another proof methodology, it is argued in [23] that the regret of PSRL grows as the square root of the state space size and not linearly. While this more recent methodology leads to a better bound, in our paper we choose to use the more conservative approach of [24] because we believe that the proof used in [23] is not correct (in particular the use of a deterministic V in Equation (16) of the proof of Lemma 3 in Appendix A in the arXiv version of [23] seems incompatible with the use of Lemma 4 of the same paper).

5.2 Minimax Lower Bound

After establishing the regret upper bound for both algorithms, a natural question is: can we do better? Or in other terms, does there exist a learning algorithm with a smaller regret. To answer this question, the metric used in the literature is the notion of minimax lower bound: for a given set of parameters (S, n, K, β), a minimax lower bound is a lower bound on the quantity $\inf_{\mathcal{L}} \sup_{\mathcal{M}} \mathrm{Reg}(K, \mathcal{L}, \mathcal{M})$, where the supremum is taken among all possible models that have parameters (S, n, K, β) and the infimum is taken over all possible learning algorithms. The next theorem provides a lower bound on the Bayesian regret. It is therefore stronger than a minimax bound for two reasons: First, the Bayesian regret is an average over models, which means that there exists at least one model that has a larger regret than the Bayesian lower bound; And second, in Theorem 3, we allow the algorithm to depend on the prior distribution ϕ and to use this information.

Theorem 3 (Lower bound)**.** *For any state size S, number of bandits n, discount factor β and number of episodes $K \geq 16S$, there exists a prior distribution ϕ on Markovian bandit problems with*

parameters (S, n, K, β) *such that, for any learning algorithm* $\mathcal{L}$*:*

$$\mathrm{BayReg}(K, \mathcal{L}, \phi) \geq \frac{1}{60}\sqrt{\frac{SnK}{(1-\beta)}}. \tag{10}$$

The proof is given in Appendix B and uses a counterexample inspired by the one of [2]. Note that for general MDPs, the minimax lower bound obtained in [2] says that a learning algorithm cannot have a regret smaller than $\Omega\big(\sqrt{\tilde{S}\tilde{A}\tilde{T}}\big)$, where $\tilde{S}$ is the number of states of the MDP, $\tilde{A}$ is the number of actions and $\tilde{T}$ is the number of time steps. Yet, the lower bound of [2] is not directly applicable to our case with $\tilde{S} = S^n$ because Markovian bandit problems are very specific instances of MDPs and this can be exploited by the learning algorithm.

Apart from the logarithmic terms, the lower bound provided by Theorem 3 differs from the bound of Theorem 1 and 2 by a factor $\sqrt{S}/(1-\beta)$. This factor is similar to the one observed for PSRL and UCRL2 [2, 24]. There are various factors that could explain this. We believe that the extra factor $1/(1-\beta)$ might be half due to the episodic nature of MB-PSRL and MB-UCRL2 (when $1/(1-\beta)$ is large, algorithms with internal episodic updates might have smaller regret) and half due to the fact that the lower bound of Theorem 3 is not optimal and could include a term $1/\sqrt{1-\beta}$ (similar to the term $O(\sqrt{D})$ of the lower bound of [2]). The factor $1/\sqrt{S}$ between our two bounds comes from our use of Weissman's inequality. It might be possible that both algorithms are not optimal with respect to this term and we believe that an algorithm inspired by UCBVI [4] could remove the term in $\sqrt{S}$. Yet this would have two prices: a higher-order term in $1/(1-\beta)$, but most importantly, a higher computational complexity, as indicated next.

6 MB-PSRL is scalable whereas MB-UCRL2 is not

In this section, we give the computational complexity of MB-PSRL and MB-UCRL2. By using Gittins index policy of sampled MDP, MB-PSRL has complexity that is linear in the number of bandits. MB-UCRL2 has exponential complexity due to extended value iteration. We conclude the section by asserting the impossibility of reducing computational complexity of OFU-based algorithms using local index computation.

6.1 Runtime analysis

MB-PSRL If one excludes the simulation of the MDP, the computational cost of each episode is essentially due to three components: Updating the observations, sampling from the posterior distribution and computing the optimal policy. The first two are relatively fast when the conjugate posterior has a closed form: updating the observation takes $O(1)$ at each time, and sampling from the posterior can be done in $O(nS^2)$ – more details on posterior distributions are given in Appendix E. When the conjugate posterior is implicit (*i.e.*, under the integral form), the computation can be higher but remains linear in the number of bandits. Finally, the computation of the Gittins indices can be done in $O(nS^3)$ as explained in Section 2.2. So, we can conclude that MB-PSRL successfully escapes from the curse of dimensionality.

MB-UCRL2 While MB-UCRL2 has a regret equivalent to the one of MB-PSRL, its computational complexity, and in particular the complexity of computing an *optimistic* policy that maximizes (6) is large. Such a policy is computed by using *extended value iteration* [2], but this computation is polynomial in the number of states of the global MDP and is therefore exponential in the number of bandits, precisely $O(nS^{2n})$. In factored MDPs – [22, 37, 27, 31] – this problem is circumvented by assuming the existence of an oracle that can solve (6).

All those OFU-based algorithms for Markovian bandits (called "MB-OFU" algorithms in the following) improve their regret bound by exploiting the structure of Markovian bandit problem. However, their computational complexity remains exponential in the number of bandits unless one can also exploit the bandit structure to compute an optimal policy solving (6) efficiently. In the following we show that unlike for MB-PSRL, such an optimal policy cannot be based on local indices for MB-OFU.

6.2 MB-OFU Algorithms Cannot use Index Policies

In order to achieve (8), OFU-based algorithms compute an optimistic MDP $\bar{\mathcal{M}}_k$ and policy π_k simultaneously. We now show that this simultaneous computation cannot be replaced by the computation of local indices for each bandit although there exists an index policy that is optimistic in the sense of (8).

More precisely, let us consider that the estimates and confidence bounds for a given bandit a are $\hat{\mathcal{B}}_a = (\hat{\boldsymbol{r}}_a, \hat{Q}_a, \epsilon_a^r, \epsilon_a^Q)$. We say that an algorithm computes indices locally for Bandit a if for each $x_a \in \mathcal{S}_a$, it computes an index $I^{\hat{\mathcal{B}}_a}(x_a)$ by using only $\hat{\mathcal{B}}_a$ but not $\hat{\mathcal{B}}_b$ for any $b \neq a$. We denote by $\pi^{I(\hat{\mathcal{B}})}$ the index policy that uses index $I^{\hat{\mathcal{B}}_a}$ for bandit a and by $\mathbb{M}(\hat{\mathcal{B}})$ the set of Markovian bandit problems $\bar{\mathcal{M}}$ that satisfy (5).

Theorem 4. *For any algorithm that computes indices locally, there exist a Markovian bandit problem $\mathcal{M}$, an initial distribution ρ and estimates $\hat{\mathcal{B}}_a = (\hat{\boldsymbol{r}}_a, \hat{Q}_a, \epsilon_a^r, \epsilon_a^Q)$ such that $\mathcal{M} \in \mathbb{M}(\hat{\mathcal{B}})$ and*

$$\sup_{\bar{\mathcal{M}} \in \mathbb{M}(\hat{\mathcal{B}})} V_{\bar{\mathcal{M}}}^{\pi^{I(\hat{\mathcal{B}})}}(\rho) < \sup_{\pi} V_{\mathcal{M}}^{\pi}(\rho).$$

Proof. The proof presented in Appendix D is obtained by constructing a set $\mathbb{M}$ and two MDPs $\mathcal{M}_1$ and $\mathcal{M}_2$ in $\mathbb{M}$ such that (8) cannot hold simultaneously for both $\mathcal{M}_1$ and $\mathcal{M}_2$. □

This theorem implies that one cannot define local indices such that (8) holds for all bandit problems $\mathcal{M} \in \mathbb{M}$. Yet, the use of this inequality is central in the regret analysis of all OFU-based reinforcement learning algorithms (see for instance the proof of UCRL2 in [2] or the proof of UCBVI, and in particular Lemma 18 of [4]). This implies that the current methodology to obtain regret bounds for OFU-based algorithms is not applicable to bound the regret of any algorithm that computes indices locally.

Note that for any set $\mathbb{M}$ such that $\mathcal{M} \in \mathbb{M}$, there exists an index policy π^{ind} that is optimistic because all MDPs in $\mathbb{M}$ are Markovian bandit problems. This optimistic index policy satisfies

$$\sup_{\bar{\mathcal{M}} \in \mathbb{M}} V_{\bar{\mathcal{M}}}^{\pi^{\mathrm{ind}}} \geq \sup_{\pi} V_{\mathcal{M}}^{\pi}.$$

This means that restricting to index policies is not a restriction for optimism. What Theorem 4 shows is that an optimistic index policy can be defined only after the most optimistic MDP $\bar{\mathcal{M}} \in \mathbb{M}$ is computed and this computation depends on the confidence sets of all bandits.

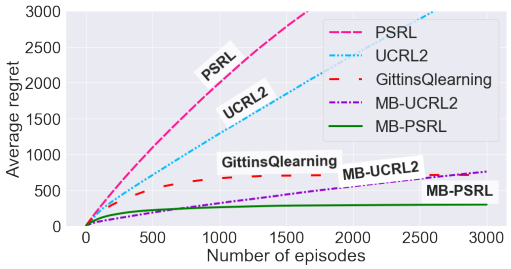
Therefore, we believe that no optimistic policy can be computed locally: they should all require the joint knowledge of all $(\hat{\mathcal{B}}_a)_{a \in [n]}$. In a dual manner, this also suggests that there is no index based OFU algorithm whose index are defined locally and that has a sub-linear regret. Although this is not a proof, this suggests that the Q-learning algorithm from [8] does not have a sub-linear regret.

7 Numerical Experiments

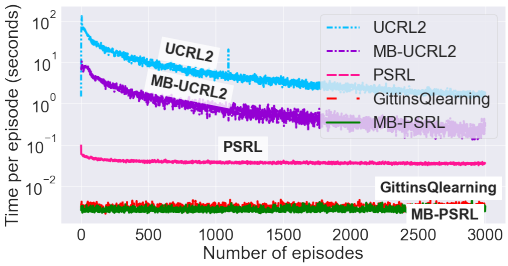
In complement to our theoretical analysis, we report, in this section, the performance of MB-PSRL and MB-UCRL2 in a model taken from the literature. We compare them with three alternative algorithms: Q-learning based presented in [8] (that we call GittinsQlearning), vanilla PSRL and UCRL2. We design an environment with 3 bandits, all following a Markov chain that is obtained by applying the optimal policy on the river swim MDP. A detailed description is given in Appendix E, along with all hyperparameters that we used.

Our numerical experiments suggest that MB-PSRL outperforms other algorithms in term of average regret and is computationally less expensive than other algorithms. To ensure reproducibility, the code of all of our experiments is available in the supplementary material.

Performance Result We investigate the average regret and policy computation time of each algorithm. To do so, we run each algorithm for 80 simulations and for $K = 3000$ episodes per simulation. We arbitrarily choose the discount factor $\beta = 0.99$. In Figure 1a, we show the average cumulative regret of the 5 algorithms. We observe that, as expected, the cumulative regrets of



(a) Average cumulative regret in function of the number of episodes.



(b) Average runtime per episode. The vertical axis is in log-scale.

Figure 1: Result from 80 simulations in a Markovian Bandits problem with three 4-state random walk chains given in Table 1. The horizontal axis is the number of episodes. Each algorithm is identified by a unique color for all figures.

PSRL and UCRL2 are larger than those of MB-PSRL and MB-UCRL2. This indicates that using an algorithm specifically designed for Markovian bandit problems is much more efficient. Moreover, we observe that MB-PSRL obtains the best performance and that its regret seems to grow slower than $O(\sqrt{K})$. This is in accordance to what was observed for PSRL in [24]. Note that the expected number of time steps after K episodes is $K/(1-\beta)$ which means that in our setting with $K = 3000$ episodes there are $300\,000$ time steps in average. In Figure 1b, we compare the computation time of the various algorithms. We observe that there are several orders of magnitude of difference between the computation time (the y-axis is in log-scale) and that MB-PSRL and GittinsQlearning, the index-based algorithms, are the fastest by far. Moreover, the computation time of these algorithms seem to be independent of the number of episodes (which is also true for PSRL). These two figures show that MB-PSRL has the smallest regret and computation time among all compared algorithms.

Robustness (Larger Models and Different Priors) To test the robustness of MB-PSRL, we conduct two more sets of experiments that are reported in Appendix F. They confirm the superiority of MB-PSRL. The first experiment is an example from [8] with 9 bandits each having 11 states. This model illustrates the effect of the curse of dimensionality: the global MDP has 11^9 states which implies that the runtime of PSRL, UCRL2 and MB-UCRL2 makes them impossible to use, while MB-PSRL and GittinsQLearning take a few minutes to complete 3000 episodes. In this example, MB-PSRL seems to converge faster to the optimal policy than GittinsQlearning (remember that GittinsQlearning has no regret guarantee). The second experiment tests the robustness of MB-PSRL and PSRL to the choice of prior distribution. We provide numerical evidences that show that, even when MB-PSRL is run with a prior ϕ that is not the one from which $\mathcal{M}$ is drawn, the regret of MB-PSRL remains acceptable (around twice the regret obtained with a correct prior).

8 Conclusion

In this paper, we study the properties of two algorithms that are tailored for Markovian bandit problem: MB-PSRL and MB-UCRL2. We show that their regret is close to the lower bound that we derive for this problem. MB-PSRL has a runtime that scales linearly with the number of bandits while the runtime of MB-UCRL2 is exponential.

We conjecture that there does not exist any natural adaptation of learning algorithms based on optimism for Markovian bandits with a sub-linear regret and a linear runtime in the number of bandits.

References

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47(2):235–256, May 2002. ISSN 1573-0565. doi: 10.1023/A:1013689704352.

[2] P. Auer, T. Jaksch, and R. Ortner. Near-optimal Regret Bounds for Reinforcement Learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 89–96. Curran Associates, Inc., 2009.

[3] K. Avrachenkov and V. S. Borkar. Whittle index based Q-learning for restless bandits with average reward. *arXiv:2004.14427 [cs, math, stat]*, Apr. 2020.

[4] M. G. Azar, I. Osband, and R. Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272, 2017.

[5] P. L. Bartlett and A. Tewari. REGAL: A regularization based algorithm for reinforcement learning in weakly communicating MDPs. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 35–42, Montreal, Quebec, Canada, June 2009. AUAI Press. ISBN 978-0-9749039-5-8.

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

[7] J. Chakravorty and A. Mahajan. Multi-Armed Bandits, Gittins Index, and its Calculation. In N. Balakrishnan, editor, *Methods and Applications of Statistics in Clinical Trials*, pages 416–435. John Wiley & Sons, Inc., Hoboken, NJ, USA, June 2014. ISBN 978-1-118-59633-3 978-1-118-30476-1. doi: 10.1002/9781118596333.ch24.

[8] M. O. Duff. Q-Learning for Bandit Problems. In A. Prieditis and S. Russell, editors, *Machine Learning Proceedings 1995*, pages 209–217. Morgan Kaufmann, San Francisco (CA), Jan. 1995. ISBN 978-1-55860-377-6. doi: 10.1016/B978-1-55860-377-6.50034-7.

[9] S. Filippi, O. Cappé, and A. Garivier. Optimism in Reinforcement Learning and Kullback-Leibler Divergence. *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 115–122, Sept. 2010. doi: 10.1109/ALLERTON.2010.5706896.

[10] D. Fink. *A Compendium of Conjugate Priors*. 1997.

[11] R. Fruit, M. Pirotta, A. Lazaric, and E. Brunskill. Regret Minimization in MDPs with Options without Prior Knowledge. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3166–3176. Curran Associates, Inc., 2017.

[12] J. Fu, Y. Nazarathy, S. Moka, and P. G. Taylor. Towards Q-learning the Whittle Index for Restless Bandits. In *2019 Australian New Zealand Control Conference (ANZCC)*, pages 249–254, Nov. 2019. doi: 10.1109/ANZCC47194.2019.8945748.

[13] J. Gittins, K. Glazebrook, and R. Weber. *Multi-Armed Bandit Allocation Indices, 2nd Edition*, volume 33. 02 2011. doi: 10.1002/9780470980033.ch8.

[14] J. C. Gittins. Bandit Processes and Dynamic Allocation Indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):148–164, Jan. 1979. ISSN 00359246. doi: 10.1111/j.2517-6161.1979.tb01068.x.

[15] A. Gopalan and S. Mannor. Thompson Sampling for Learning Parameterized Markov Decision Processes. In *Conference on Learning Theory*, pages 861–898. PMLR, June 2015.

[16] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, 19(1):399–468, Oct. 2003. ISSN 1076-9757.

[17] T. Jaakkola, M. I. Jordan, and S. P. Singh. Convergence of Stochastic Iterative Dynamic Programming Algorithms. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 703–710. Morgan-Kaufmann, 1994.

[18] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan. Is Q-Learning Provably Efficient? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4863–4873. Curran Associates, Inc., 2018.

[19] M. N. Katehakis and A. F. Veinott. The Multi-Armed Bandit Problem: Decomposition and Computation. *Mathematics of Operations Research*, 12(2):262–268, May 1987. ISSN 0364-765X. doi: 10.1287/moor.12.2.262.

[20] M. L. Littman and C. Szepesvári. A Generalized Reinforcement-Learning Model: Convergence and Applications. Technical Report, Brown University, USA, 1996.

[21] K. P. Murphy. Conjugate bayesian analysis of the gaussian distribution. Technical report, 2007.

[22] I. Osband and B. V. Roy. Near-optimal reinforcement learning in factored MDPs. In *Proc. of the 27th Int. Conf. on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 604–612, Montreal, Canada, Dec. 2014. MIT Press.

[23] I. Osband and B. Van Roy. Why is posterior sampling better than optimism for reinforcement learning? In *Int. Conf. on Machine Learning*, pages 2701–2710. PMLR, 2017.

[24] I. Osband, D. Russo, and B. Van Roy. (More) Efficient Reinforcement Learning via Posterior Sampling. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3003–3011. Curran Associates, Inc., 2013.

[25] Y. Ouyang, M. Gagrani, A. Nayyar, and R. Jain. Learning Unknown Markov Decision Processes: A Thompson Sampling Approach. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1333–1342. Curran Associates, Inc., 2017.

[26] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1st edition, 1994. ISBN 978-0-471-61977-2.

[27] A. Rosenberg and Y. Mansour. Oracle-Efficient Reinforcement Learning in Factored MDPs with Unknown Structure. *arXiv:2009.05986 [cs, stat]*, Sept. 2020.

[28] D. J. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

[29] M. J. A. Strens. A Bayesian Framework for Reinforcement Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 943–950, San Francisco, CA, USA, June 2000. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-707-1.

[30] W. 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. ISSN 0006-3444. doi: 10.2307/2332286.

[31] Y. Tian, J. Qian, and S. Sra. Towards Minimax Optimal Reinforcement Learning in Factored Markov Decision Processes. *arXiv:2006.13405 [cs, math, stat]*, June 2020.

[32] J. N. Tsitsiklis. Asynchronous Stochastic Approximation and Q-Learning. *Machine Learning*, 16(3):185–202, Sept. 1994. ISSN 1573-0565. doi: 10.1023/A:1022689125041.

[33] R. Vershynin. *High-dimensional Probability*. Cambridge University Press, 2018.

[34] R. Weber. On the Gittins Index for Multiarmed Bandits. *Annals of Applied Probability*, 2(4): 1024–1033, Nov. 1992. ISSN 1050-5164, 2168-8737. doi: 10.1214/aoap/1177005588.

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

[36] P. Whittle. *Optimal Control: Basics and Beyond*. Wiley Interscience Series in Systems and Optimization. Wiley, 1996. ISBN 9780471960997. URL `https://books.google.fr/books?id=6wZhQgAACAAJ`.

[37] Z. Xu and A. Tewari. Reinforcement Learning in Factored MDPs: Oracle-Efficient Algorithms and Tighter Regret Bounds for the Non-Episodic Setting. *arXiv:2002.02302 [cs, stat]*, June 2020.

[38] A. Zanette and E. Brunskill. Problem Dependent Reinforcement Learning Bounds Which Can Identify Bandit Structure in MDPs. In *International Conference on Machine Learning*, pages 5747–5755. PMLR, July 2018.

Appendix

A Proof of Theorem 1

A.1 Overview of the Proof

This proof of Theorem 1 is based on multiple technical lemmas that are stated and proven below. In this section, we show how those lemmas are articulated to prove the main result.

Recall that π_* is the optimal policy of the original MDP $\mathcal{M}$ and that π_k is the policy that is applied for episode k. π_k is optimal for the MDP $\mathcal{M}_k$ that is drawn before episode k. By definition, the Bayesian regret of the MB-PSRL is

$$\begin{aligned}\mathrm{BayReg}(K, \text{MB-PSRL}, \phi) &= \sum_{k=1}^{K} \mathbb{E}\left[V_{\mathcal{M}}^{\pi_*}(\boldsymbol{X}_{t_k}) - V_{\mathcal{M}}^{\pi_k}(\boldsymbol{X}_{t_k})\right] \\ &= \sum_{k=1}^{K} \Big(\mathbb{E}\left[V_{\mathcal{M}}^{\pi_*}(\boldsymbol{X}_{t_k}) - V_{\mathcal{M}_k}^{\pi_k}(\boldsymbol{X}_{t_k})\right] + \mathbb{E}\left[V_{\mathcal{M}_k}^{\pi_k}(\boldsymbol{X}_{t_k}) - V_{\mathcal{M}}^{\pi_k}(\boldsymbol{X}_{t_k})\right] \Big),\end{aligned} \tag{11}$$

where the expectation includes the random evolution of $\boldsymbol{X}_t$ when running the algorithm, the random draws of $\mathcal{M}_k$ done at the beginning of each episode, and also the random draw of $\mathcal{M}$ (recall that in our Bayesian model, $\mathcal{M}$ is drawn according to the prior distribution).

Equation (11) is the sum of two terms. We prove in Lemma 1 that the first term is 0. This implies that to bound the regret, it suffices to compare the performance of π_k in the original MDP $\mathcal{M}$ and in the MDP $\mathcal{M}_k$ that is drawn at the start of episode k. To do so, we show in Lemma 2 that

$$\mathbb{E}\left[V_{\mathcal{M}_k}^{\pi_k}(\boldsymbol{X}_{t_k}) - V_{\mathcal{M}}^{\pi_k}(\boldsymbol{X}_{t_k})\right] \leq \mathbb{E}\Bigg[\sum_{t=t_k}^{t_{k+1}-1} |r_k(X_{t,A_t}) - r(X_{t,A_t})| + H_k \left\| Q_k(X_{t,A_t}, \cdot) - Q(X_{t,A_t}, \cdot) \right\|_1 \Bigg] \tag{12}$$

where $\|Q_k(x_a, \cdot) - Q(x_a, \cdot)\|_1 = \sum_{y_a} |Q_k(x_a, y_a) - Q(x_a, y_a)|$.

For a bandit a and a state $x_a \in \mathcal{S}_a$, we denote[2] by $N_k(x_a) = \sum_{t=0}^{t_k - 1} \mathbb{I}_{\{X_{t,A_t} = x_a\}}$ the number of times that Bandit a is activated before episode k while being in state x_a. Equation (12) relates the performance gap to the distance between the reward functions and transition matrices of the MDPs $\mathcal{M}$ and $\mathcal{M}_k$. We show in Lemma 3 that $r_k(x_a)$ and $Q_k(x_a, \cdot)$ are good estimations of the true values $r(x_a)$ and $Q(x_a, \cdot)$ as soon as $N_k(x_a)$ is large enough. More precisely, let $C_k = \sqrt{2\log(8SnKt_k)}$. With probability at least $1 - 1/K$, for all a, x_a and $k \geq 1$,

$$|r_k(x_a) - r(x_a)| \leq \frac{C_k}{\sqrt{\max(1, N_k(x_a))}} \text{ and } \|Q_k(x_a, \cdot) - Q(x_a, .)\|_1 \leq \frac{2C_k + 3\sqrt{S}}{\sqrt{\max(1, N_k(x_a))}}. \tag{13}$$

Combining this with Equation (11) and (12) shows that

$$\mathrm{BayReg}(K, \text{MB-PSRL}, \phi) \leq \frac{1}{(1-\beta)} + \mathbb{E}\left[\sum_{k=1}^{K} \sum_{t=t_k}^{t_{k+1}-1} \frac{C_k + (2C_k + 3\sqrt{S})H_k}{\sqrt{\max(1, N_k(X_{t,A_t}))}} \right], \tag{14}$$

where the term $\frac{1}{(1-\beta)}$ comes from the fact that (13) is not true with probability at most $1/K$ (see Equation (26) for more details).

The analysis of the right term of (14) is more tricky and uses the fact that there cannot be too many large terms in this sum because if a bandit is activated many times, then $1/\sqrt{N_k(X_{t,A_t})}$ is small. We provide a detailed analysis of this sum in Appendix A.5 which concludes the proof. The main technical hurdle here is to deal with the K random episodes $H_1, \ldots, H_k$. This is specific to our

[2] In the paper, we use the notation $\mathbb{I}_{\{E\}}$ to denote a random variable that equals 1 if E is true and 0 otherwise. For instance, $\mathbb{I}_{\{Y_i = y\}} = 1$ if $Y_i = y$ and 0 otherwise.

approach compared to the analysis of finite horizons. To bound this, one needs to bound terms of the form $\mathbb{E}\left[\max_{1\le k\le K}(H_k)^\alpha \log H_k\right]$ with $\alpha \in \{1.5, 2\}$ (see Equation (29)). To bound this, we first use that $\log(x) \le x^\epsilon/(e\epsilon)$ (Lemma 4) to remove the logarithmic term and then use the geometric distribution of H_k to show that $\mathbb{E}\left[\max_{1\le k\le K}(H_k)^{\alpha+\epsilon}\right] = O((\frac{\log K}{1-\beta})^{\alpha+\epsilon})$ (see Lemma 5).

A.2 Expectation Identity

Lemma 1. *Assume that the MDP $\mathcal{M}$ is drawn according to the prior ϕ and that $\mathcal{M}_k$ is draw according to the posterior $\phi(\cdot \mid \mathcal{O}_{k-1})$. Then, for any $\mathcal{O}_{k-1}$-measurable function g, one has:*

$$\mathbb{E}\left[g(\mathcal{M})\right] = \mathbb{E}\left[g(\mathcal{M}_k)\right]. \tag{15}$$

Proof. At the start of each episode k, MB-PSRL computes the posterior distribution of $\mathcal{M}$ conditioned on the observations $\mathcal{O}_{k-1}$, and draws $\mathcal{M}_k$ from it. This implies that $\mathcal{M}$ and $\mathcal{M}_k$ are identically distributed conditioned on $\mathcal{O}_{k-1}$. Consequently, if g is a $\mathcal{O}_{k-1}$-measurable function, one has:

$$\mathbb{E}\left[g(\mathcal{M}) \mid \mathcal{O}_{k-1}\right] = \mathbb{E}\left[g(\mathcal{M}_k) \mid \mathcal{O}_{k-1}\right].$$

Equation (15) then follows from the tower rule. □

Note that in particular, the above lemma implies that $\mathbb{E}\left[V^{\pi_*}_{\mathcal{M}}(\boldsymbol{X}_{t_k})\right] = \mathbb{E}\left[V^{\pi_k}_{\mathcal{M}_k}(\boldsymbol{X}_{t_k})\right]$ because π_k is $\mathcal{O}_{k-1}$-measurable.

A.3 Regret Decomposition

Lemma 2. *The Bayesian regret at episode k satisfies:*

$$\mathbb{E}\left[V^{\pi_k}_{\mathcal{M}_k}(\boldsymbol{X}_{t_k}) - V^{\pi_k}_{\mathcal{M}}(\boldsymbol{X}_{t_k})\right] \le \mathbb{E}\left[\sum_{t=t_k}^{t_{k+1}-1} \left(|r_k(X_{t,A_t}) - r(X_{t,A_t})| + H_k \left\|Q_k(X_{t,A_t},\cdot) - Q(X_{t,A_t},\cdot)\right\|_1\right)\right]. \tag{16}$$

Proof. Let us define as $W^{\pi_k}_{\mathcal{M},H}(\boldsymbol{x})$ the reward of the MDP $\mathcal{M}$ that starts in state $\boldsymbol{x}$ and lasts for H time steps:

$$W^{\pi_k}_{\mathcal{M},H}(\boldsymbol{x}) = \mathbb{E}\left[\sum_{t=0}^{H-1} r(X_{t,A_t}) \mid \boldsymbol{X}_0 = \boldsymbol{x}, H, A_t = \pi_k(\boldsymbol{X}_t)\right].$$

So, $V^{\pi_k}_{\mathcal{M}}(\boldsymbol{x}) = \mathbb{E}[W^{\pi_k}_{\mathcal{M},H}(\boldsymbol{x}) \mid H \sim \mathrm{Geom}(1-\beta)]$. By applying one step of value iteration, and denoting $a = \pi_k(\boldsymbol{x})$ for clarity, we have that

$$\begin{aligned} W^{\pi_k}_{\mathcal{M},H}(\boldsymbol{x}) &= r(x_a) + \mathbb{E}\left[W^{\pi_k}_{\mathcal{M},H-1}(\boldsymbol{X}_1) \mid \boldsymbol{X}_0 = \boldsymbol{x}, H, A_0 = a\right] \\ &= r(x_a) + \sum_{\boldsymbol{y}} P(\boldsymbol{x}, a, \boldsymbol{y}) W^{\pi_k}_{\mathcal{M},H-1}(\boldsymbol{y}). \end{aligned} \tag{17}$$

Comparing the sampled MDP $\mathcal{M}_k$ with the original $\mathcal{M}$ and using (17), one has

$$W^{\pi_k}_{\mathcal{M}_k,H}(\boldsymbol{x}) - W^{\pi_k}_{\mathcal{M},H}(\boldsymbol{x}) = r_k(x_a) - r(x_a) + \sum_{\boldsymbol{y}} P_k(\boldsymbol{x}, a, \boldsymbol{y}) W^{\pi_k}_{\mathcal{M}_k,H-1}(\boldsymbol{y}) - \sum_{\boldsymbol{y}} P(\boldsymbol{x}, a, \boldsymbol{y}) W^{\pi_k}_{\mathcal{M},H-1}(\boldsymbol{y}).$$

Note that in the above equation, the last term is of the form $P_k W^{\pi_k}_{\mathcal{M}_k} - P W^{\pi_k}_{\mathcal{M}}$, which is equal to $(P_k - P) W^{\pi_k}_{\mathcal{M}_k} + P(W^{\pi_k}_{\mathcal{M}_k} - W^{\pi_k}_{\mathcal{M}})$. Moreover, as $r(x_a) \le 1$, it should be clear that $W^{\pi_k}_{\mathcal{M},H-1}(\boldsymbol{y})$ and $W^{\pi_k}_{\mathcal{M}_k,H-1}(\boldsymbol{y})$ are both smaller[3] than H. Plugging this to the above equation shows that:

$$\begin{aligned} \left|W^{\pi_k}_{\mathcal{M}_k,H}(\boldsymbol{x}) - W^{\pi_k}_{\mathcal{M},H}(\boldsymbol{x})\right| &\le |r_k(x_a) - r(x_a)| + H \sum_{\boldsymbol{y}} |P_k(\boldsymbol{x}, a, \boldsymbol{y}) - P(\boldsymbol{x}, a, \boldsymbol{y})| \\ &\quad + \sum_{\boldsymbol{y}} P(\boldsymbol{x}, a, \boldsymbol{y})(W^{\pi_k}_{\mathcal{M}_k,H-1}(\boldsymbol{y}) - W^{\pi_k}_{\mathcal{M},H-1}(\boldsymbol{y})) \\ &= |r_k(x_a) - r(x_a)| + H \left\|P_k(\boldsymbol{x}, a, \cdot) - P(\boldsymbol{x}, a, \cdot)\right\|_1 \\ &\quad + \mathbb{E}\left[W^{\pi_k}_{\mathcal{M}_k,H-1}(\boldsymbol{X}_1) - W^{\pi_k}_{\mathcal{M},H-1}(\boldsymbol{X}_1) \mid \boldsymbol{X}_0 = \boldsymbol{x}, H, A_0 = a, \mathcal{M}, \mathcal{M}_k, \pi_k\right]. \end{aligned}$$

[3] In fact, $W^{\pi_k}_{\mathcal{M},H-1}(\boldsymbol{y}) \le H - 1$. We bound this quantity by H to simplify the expression.

Note that in the equation above, the only random variable is $\boldsymbol{X}_1 \sim P(\boldsymbol{x}, a, \cdot)$.

As only bandit a makes a transition, we have $\|P_k(\boldsymbol{x}, a, \cdot) - P(\boldsymbol{x}, a, \cdot)\|_1 = \|Q_k(x_a, \cdot) - Q(x_a, \cdot)\|_1$. Hence, a direct induction shows that (16) holds. □

A.4 Concentration Argument

Lemma 3. *Let $C_k = \sqrt{2\log(8SnKt_k)}$. Then, the event*

$$\mathcal{E}_k = \Bigg\{ \forall a \in [n], x_a \in \mathcal{S}_a, k' \leq k : |r_{k'}(x_a) - r(x_a)| \leq \frac{C_k}{\sqrt{\max(1, N_{k'}(x_a))}} \tag{18}$$

$$\text{and } \|Q_{k'}(x_a, \cdot) - Q(x_a, .)\|_1 \leq \frac{2C_k + 3\sqrt{S}}{\sqrt{\max(1, N_{k'}(x_a))}} \Bigg\} \tag{19}$$

is true with probability at least $1 - 1/K$.

Proof. We design confidence sets similar to [2, 5].

At the start of episode k, $\forall a \in [n]$, let $N_k(x_a) = \sum_{t=0}^{t_k - 1} \mathbb{I}_{\{X_{t,A_t} = x_a\}}$ be the number of times so far that a bandit a was activated in state x_a. Assume that $N_k(x_a) \geq 1$ (if $N_k(x_a) = 0$, the bound of (18)-(19) is trivial). Let $\hat{r}_k$ and $\hat{Q}_k$ be the empirical mean reward vector and transition matrix. In particular, $\hat{r}_k(x_a)$ is the empirical mean reward earned when bandit a is chosen while being in state x_a:

$$\hat{r}_k(x_a) = \frac{1}{N_k(x_a)} \sum_{t=0}^{t_k - 1} R_t \mathbb{I}_{\{A_t = a \wedge X_{t,A_t} = x_a\}},$$

and $\hat{Q}_k(x_a, y_a)$ is the fraction of times that bandit a moved from x_a to y_a:

$$\hat{Q}_k(x_a, y_a) = \frac{1}{N_k(x_a)} \sum_{t=0}^{t_k - 1} \mathbb{I}_{\{A_t = a \wedge X_{t,A_t} = x_a \wedge X_{t+1,A_t} = y_a\}}.$$

By Hoeffding's inequality, for any $\epsilon > 0$, one has:

$$\mathbb{P}\left(|\hat{r}_k(x_a) - r(x_a)| \geq \epsilon \mid N_k(x_a)\right) \leq 2e^{-2N_k(x_a)\epsilon^2}.$$

In particular, this holds for $\epsilon = \sqrt{\frac{\log(8Snt_k/\delta)}{2N_k(x_a)}}$ where $0 < \delta < 1$. As $N_k(x_a) \leq t_k$, by using the union-bound, this implies that:

$$\begin{aligned} &\mathbb{P}\left(\forall a, x_a, k' \leq k : |\hat{r}_{k'}(x_a) - r(x_a)| \geq \sqrt{\frac{\log(8Snt_k/\delta)}{2N_{k'}(x_a)}}\right) \\ &\quad \leq \sum_{t=0}^{t_{k'}-1} \sum_a \sum_{x_a} \mathbb{P}\left(|\hat{r}_{k'}(x_a) - r(x_a)| \geq \sqrt{\frac{\log(8Snt_k/\delta)}{2N_{k'}(x_a)}} \mid N_{k'}(x_a) = t\right) \\ &\quad \leq 2t_k nS e^{-2N_{k'}(x_a)\frac{\log(8Snt_k/\delta)}{2N_{k'}(x_a)}} = \frac{\delta}{4}. \end{aligned} \tag{20}$$

The above equation shows that $\hat{r}_k$ is close to the reward of the MDP $\mathcal{M}$. Note that the probability expressed in (20) only depends on events before episode k (and hence on $\mathcal{O}_{k-1}$). So, it can be expressed as the expectation of an $\mathcal{O}_{k-1}$-measurable function. Hence, Lemma 1 implies that Equation (20) holds if r is replaced by r_k:

$$\mathbb{P}\left(\forall a, x_a, k' \leq k : |\hat{r}_{k'}(x_a) - r_{k'}(x_a)| \geq \sqrt{\frac{\log(8Snt_k/\delta)}{2N_{k'}(x_a)}}\right) \leq \frac{\delta}{4}. \tag{21}$$

By using the union bound of the type $\mathbb{P}\left(|r_k - r| \geq 2\epsilon\right) \leq \mathbb{P}\left(|\hat{r}_k - r| \geq \epsilon\right) + \mathbb{P}\left(|\hat{r}_k - r_k| \geq \epsilon\right)$ and let $\delta = 1/K$, this implies that

$$\mathbb{P}\left(\forall a, x_a, k' \leq k : |r_{k'}(x_a) - r(x_a)| \geq \sqrt{\frac{2\log(8SnKt_k)}{N_{k'}(x_a)}}\right) \leq \frac{1}{2K}. \tag{22}$$

The bound for Q is similar but by using Weissman's inequality [35] instead of Hoeffding's bound. Indeed, by using Equation (8) in Theorem 2.1 of [35], one has

$$\mathbb{P}\left(\left\|\hat{Q}_k(x_a, \cdot) - Q(x_a, \cdot)\right\|_1 \geq \epsilon\right) \leq 2^S e^{-N_k(x_a)\epsilon^2/2}.$$

Following the same approach as for Equation (20) with $\epsilon = \sqrt{2\log(4Snt_k 2^S/\delta)/N_k(x_a)}$, we have:

$$\begin{aligned}\mathbb{P}\left(\forall a, x_a, k' \leq k : \left\|\hat{Q}_{k'}(x_a, \cdot) - Q(x_a, \cdot)\right\|_1 \geq \sqrt{\frac{2\log(4Snt_k 2^S/\delta)}{N_{k'}(x_a)}}\right) \\ \leq 2^S t_k n S e^{-N_{k'}(x_a)\frac{2\log(4Snt_k 2^S/\delta)}{2N_{k'}(x_a)}} = \frac{\delta}{4}.\end{aligned} \tag{23}$$

Let $\delta = 1/K$. By definition of $C_k = \sqrt{2\log(8SnKt_k)}$ and since $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$, we have $\sqrt{2\log(4SnKt_k 2^S)} = \sqrt{2\log(8SnKt_k) + 2(S-1)\log 2} \leq C_k + \sqrt{2(S-1)\log 2} \leq C_k + 1.5\sqrt{S}$. Hence:

$$\mathbb{P}\left(\forall a, x_a, k' \leq k : \left\|\hat{Q}_{k'}(x_a, \cdot) - Q(x_a, \cdot)\right\|_1 \geq \frac{C_k + 1.5\sqrt{S}}{\sqrt{N_{k'}(x_a)}}\right) \leq \frac{1}{4K}.$$

As for r, by Lemma 1, the above equation also holds when replacing Q by Q_k. Hence, by using the union bound one has:

$$\mathbb{P}\left(\forall a, x_a, k' \leq k : \left\|Q_{k'}(x_a, \cdot) - Q(x_a, \cdot)\right\|_1 \geq \frac{2C_k + 3\sqrt{S}}{\sqrt{N_{k'}(x_a)}}\right) \leq \frac{1}{2K}. \tag{24}$$

The complement of the event $\mathcal{E}_k$ defined in the statement of Lemma 3 is the union of (22) and (24). Hence the union bound concludes the proof of the lemma. □

A.5 Analysis of the Sum

Recall that

$$\mathbb{E}\left[V^{\pi_k}_{\mathcal{M}_k}(\boldsymbol{X}_{t_k}) - V^{\pi_k}_{\mathcal{M}}(\boldsymbol{X}_{t_k})\right] \leq \mathbb{E}\left[\sum_{t=t_k}^{t_{k+1}-1}\left(|r_k(X_{t,A_t}) - r(X_{t,A_t})| + \left\|Q_k(X_{t,A_t}, \cdot) - Q(X_{t,A_t}, \cdot)\right\|_1 H_k\right)\right],$$

and let Δ_k be the term inside the expectation for the right side of the above equation. We have:

$$\begin{aligned}\mathbb{E}\left[\Delta_k\right] &= \mathbb{E}\left[\Delta_k \mathbb{I}_{\{\neg\mathcal{E}_k\}} + \Delta_k \mathbb{I}_{\{\mathcal{E}_k\}}\right] \\ &\leq \mathbb{E}\left[H_k\right]\mathbb{P}\left(\neg\mathcal{E}_k\right) + \mathbb{E}\left[\Delta_k \mathbb{I}_{\{\mathcal{E}_k\}}\right],\end{aligned} \tag{25}$$

where the above inequality is true because H_k is independent of $\mathcal{E}_k$ (since $\mathcal{E}_k$ is $\mathcal{O}_{k-1}$-measurable).

The first term of (25) is easy to analyze. Recall that by Lemma 3, $\mathbb{P}\left(\neg\mathcal{E}_k\right) \leq 1/K$. As $\mathbb{E}\left[H_k\right] = 1/(1-\beta)$, this implies that

$$\sum_{k=1}^{K}\mathbb{E}\left[H_k\right]\mathbb{P}\left(\neg\mathcal{E}_k\right) \leq \frac{1}{1-\beta}. \tag{26}$$

Analyzing the second term of (25) is more challenging. By definition of $\mathcal{E}_k$, this implies that

$$\begin{aligned}
\sum_{k=1}^{K} \mathbb{E}\left[\Delta_k \mathbb{I}_{\{\mathcal{E}_k\}}\right] &\leq \mathbb{E}\left[\sum_{k=1}^{K}\sum_{t=t_k}^{t_{k+1}-1} \frac{C_k + (2C_k + 3\sqrt{S})H_k}{\sqrt{\max(1, N_k(X_{t,A_t}))}}\right] \\
&\leq \mathbb{E}\left[(C_K + (2C_K + 3\sqrt{S}) \max_{k\leq K} H_k) \sum_{k=1}^{K}\sum_{t=t_k}^{t_{k+1}-1} \frac{1}{\sqrt{\max(1, N_k(X_{t,A_t}))}}\right] \quad (27) \\
&\leq \mathbb{E}\left[3(C_K + \sqrt{S}) \max_{k\leq K} H_k \sum_{k=1}^{K}\sum_{t=t_k}^{t_{k+1}-1} \frac{1}{\sqrt{\max(1, N_k(X_{t,A_t}))}}\right], \quad (28)
\end{aligned}$$

where the last inequality holds because $C_k \leq C_K$ and $\max_{k\leq K} H_k \geq 1$. Note that using (28) instead of (27) leads to a slightly worst bound but simplifies the expression.

Let $\tilde{N}_t(x_a)$ be the number of times that bandit a has been activated before time t while being in state x_a. By definition, $\tilde{N}_{t_k}(x_a) = N_k(x_a)$. Moreover, if $t \in \{t_k, \dots, t_{k+1} - 1\}$, then $\tilde{N}_t(x_a) \leq N_k(x_a) + H_k$. This shows that

$$\begin{aligned}
\sum_{k=1}^{K}\sum_{t=t_k}^{t_{k+1}-1} \frac{1}{\sqrt{\max(1, N_k(X_{t,A_t}))}} &\leq \sum_{k=1}^{K}\sum_{t=t_k}^{t_{k+1}-1} \frac{1}{\sqrt{\max(1, \tilde{N}_t(X_{t,A_t}) - H_k)}} \\
&\leq \sum_{t=0}^{t_K-1} \frac{1}{\sqrt{\max(1, \tilde{N}_t(X_{t,A_t}) - \max_k H_k)}}.
\end{aligned}$$

The above sum can be reordered to group terms by state: The above sum equals

$$\begin{aligned}
\sum_{a,x_a}\sum_{m=1}^{\tilde{N}_{t_K}(x_a)} \frac{1}{\sqrt{\max(1, m - \max_k H_k)}} &\leq \sum_{a,x_a}\left[\max_k H_k + \sum_{m=1}^{\max(1,\tilde{N}_{t_K}(x_a) - \max_k H_k)} \frac{1}{\sqrt{m}}\right], \\
&\leq Sn \max_k H_k + \sum_{a,x_a}\sum_{m=1}^{\tilde{N}_{t_K}(x_a)} \frac{1}{\sqrt{m}}, \\
&\leq Sn \max_k H_k + 2\sum_{a,x_a} \sqrt{\tilde{N}_{t_K}(x_a)},
\end{aligned}$$

where the last inequality holds because $\sum_{m=1}^{T} 1/\sqrt{m} \leq \int_1^T 1/\sqrt{x}dx \leq 2\sqrt{T}$.

Now, by Cauchy-Schwartz inequality, and because $\sum_{a,x_a} \tilde{N}_{t_K}(x_a) = t_K$, we have:

$$\sum_{a,x_a} \sqrt{\tilde{N}_{t_K}(x_a)} \leq \Big(\sum_{a,x_a} \tilde{N}_{t_K}(x_a)\Big)^{1/2} \Big(\sum_{a,x_a} 1\Big)^{1/2} = \sqrt{Snt_K}.$$

Recall that $\mathbb{P}(\neg\mathcal{E}_k) \leq 1/K$ and $C_K = \sqrt{2\log(8SnKt_K)}$. Hence, combining (25), (26), (28) and the above result shows that:

$$\mathrm{BayReg}(K, \text{MB-PSRL}, \phi) \leq \frac{1}{1-\beta} + \mathbb{E}\left[3(\sqrt{2\log(8SnKt_K)} + \sqrt{S}) \max_{k\leq K} H_k (Sn \max_{k\leq K} H_k + 2\sqrt{Snt_K})\right].$$

It should be clear that $t_K \leq K \max_{k\leq K} H_k$. Hence, denoting $H_{(K)} = \max_{k\leq K} H_k$, the second term of the above equation is smaller than

$$\begin{aligned}
&\mathbb{E}\left[3\left(\sqrt{2\log(8SnK^2)} + \sqrt{S} + \sqrt{2\log H_{(K)}}\right) H_{(K)}(SnH_{(K)} + 2\sqrt{SnKH_{(K)}})\right] \\
&\leq \mathbb{E}\left[(C' + 3\sqrt{2}\log H_{(K)})\left(SnH_{(K)}^2 + 2\sqrt{SnK}H_{(K)}^{3/2}\right)\right] \quad (29)
\end{aligned}$$

where $C' = 3\left(\sqrt{2\log(8SnK^2)} + \sqrt{S}\right)$.

By Lemma 4, for all $\epsilon > 0$, $x \geq 1$: $\log(x) \leq x^\epsilon/(e\epsilon)$. Hence, (29) is smaller than

$$(C' + \frac{3\sqrt{2}}{e\epsilon}) \left(Sn\mathbb{E}\left[H_{(K)}^{2+\epsilon}\right] + 2\sqrt{SnK}\mathbb{E}\left[H_{(K)}^{3/2+\epsilon}\right]\right)$$

which by Lemma 5 is smaller than:

$$(C' + \frac{3\sqrt{2}}{e\epsilon}) \left(5Sn + 5Sn\left(\frac{\log K}{1-\beta}\right)^{2+\epsilon} + 10\sqrt{SnK} + 10\sqrt{SnK}\left(\frac{\log K}{1-\beta}\right)^{3/2+\epsilon}\right).$$

This implies that there exists a constant C independent of all problem's parameters such that:

$$\mathrm{BayReg}(K, \text{MB-PSRL}, \phi) \leq C(\sqrt{\log(SnK)} + \sqrt{S} + \frac{1}{\epsilon}) \left(Sn\left(\frac{\log K}{1-\beta}\right)^{2+\epsilon} + \sqrt{SnK}\left(\frac{\log K}{1-\beta}\right)^{3/2+\epsilon}\right).$$

Lemma 4. *Let $\epsilon > 0$. Then, for all $x > 1$:*

$$\frac{\log x}{x^\epsilon} \leq \frac{1}{e\epsilon}.$$

Proof. Let $f : [1, \infty) \to \mathbb{R}^+]$ be the function defined by $f(x) = \frac{\log x}{x^\epsilon}$. The derivative of f is $f'(x) = (x^{\epsilon-1} - \epsilon x^{\epsilon-1}\log(x))/(\log(x))^2$. Hence, $f'(x) = 0$ if and only if $x = e^{1/\epsilon}$. This shows that f is increasing up to $f(e^{1/\epsilon}) = 1/(e\epsilon)$ and then decreasing. □

Lemma 5. *Let $\alpha \in [1, 2.5]$. Then,*

$$\mathbb{E}\left[\max_{k\leq K}(H_k)^\alpha\right] \leq 5 + 5\left(\frac{\log K}{1-\beta}\right)^\alpha. \tag{30}$$

Proof. By definition, we have

$$\begin{aligned}
\mathbb{E}\left[\max_{k\leq K}(H_k)^\alpha\right] &= \sum_{i=1}^{\infty} \mathbb{P}\left(\max_{k\leq K}(H_k)^\alpha \geq i\right) \\
&\leq \sum_{i=1}^{\infty} \min(1, K\mathbb{P}\left((H_k)^\alpha \geq i\right)) \\
&= \sum_{i=1}^{\infty} \min(1, K\beta^{i^{1/\alpha}}),
\end{aligned}$$

where the inequality comes from the union bound and the last equality is because the random variables H_k are geometrically distributed.

Let $A = \min\{i : K\beta^{i^{1/\alpha}} \leq 1\}$. Decomposing the above sum by group of size A, we have

$$\begin{aligned}
\sum_{i=1}^{\infty} \min(1, K\beta^{i^{1/\alpha}}) &= \sum_{j=0}^{\infty} \sum_{i=Aj+1}^{A(j+1)} \min(1, K\beta^{i^{1/\alpha}}) \\
&\leq \sum_{j=0}^{\infty} A\min(1, K\beta^{(Aj)^{1/\alpha}}) \\
&= A + A\sum_{j=1}^{\infty} K(\beta^{A^{1/\alpha}})^{j^{1/\alpha}},
\end{aligned} \tag{31}$$

where the inequality holds because $\beta^{i^{1/\alpha}}$ is decreasing in i.

By definition of A, we have $\beta^{A^{1/\alpha}} \leq 1/K$. This implies that the second term of Equation (31) is smaller than $\sum_{j=1}^{\infty} K(1/K)^{j^{1/\alpha}} = \sum_{j=1}^{\infty} K^{1-j^{1/\alpha}}$. As $\alpha \leq 2.5$, if $K \geq 5$, this is smaller than $\sum_{j=1}^{\infty} 5^{1-j^{1/2.5}} \approx 3.92 < 4$.

This shows that for $K \geq 5$, we have:

$$\mathbb{E}\left[\max_{k\leq K}(H_k)^\alpha\right] \leq 5A,$$

where $A = \lceil(-\log K/\log\beta)^\alpha\rceil \leq 1 + (\log K/(1-\beta))^\alpha$.

As for the case where $K \leq 4$, we have $\mathbb{E}\left[\max_{k\leq K}(H_k)^\alpha\right] \leq K\mathbb{E}\left[H_1^\alpha\right] \leq \frac{K}{(1-\beta)^\alpha}$. This term is smaller than (30) for $K \leq 4$. □

B Proof of Theorem 3

To prove the lower bound, we consider a specific Markovian bandit problem that is composed of S independent *stochastic bandit problems*. This allows us to reuse the existing minimax lower bound for stochastic bandit problems. This existing result can be stated as follows: let $\mathcal{L}^{\text{stoc.pb}}$ be a learning algorithm for the stochastic bandit problem. It is shown in Theorem 3.1 of [6] that for any number of bandits n and any number of time steps τ, there exists parameters for a stochastic bandit problem $\mathcal{M}^{\text{stoc.pb}}$ with n bandits such that the regret of the learning algorithm over τ time steps is at least $(1/20)\sqrt{n\tau}$.

$$\text{Reg}^{\text{stoc.pb}}(\tau, \mathcal{L}^{\text{stoc.pb}}, \mathcal{M}^{\text{stoc.pb}}) \geq \frac{1}{20}\sqrt{n\tau}. \tag{32}$$

This lower bound (Theorem 3.1 of [6]) is constructed by considering n stochastic bandit problems $\mathcal{M}^{\text{stoc.pb},j}$ for $j \in [n]$ with parameters that depend on τ and n. In the problem $\mathcal{M}^{\text{stoc.pb},j}$, all bandits have a reward $\gamma(\tau, n)$ except bandit j that has a reward $\gamma'(\tau, n) > \gamma(\tau, n)$. It is shown in Theorem 3.1 of [6] that a learning algorithm cannot perform uniformly well on all problems because it is impossible to distinguish them *a priori*. More precisely, in the proof of Lemma 3.2 of [6], it is shown that if the best bandit is chosen at random, then the expected (Bayesian) regret of any learning algorithm is at least $(1/20)\sqrt{n\tau}$.

As for our problem, let K be a number of episodes, β a discount factor, n a number of bandits, S a number of states per bandit and set $\tau = K/(2S(1-\beta))$. We consider a random Markovian bandit model $\mathcal{M}$ constructed as follows. Each bandit a has S states with the state space $\mathcal{S}_a = \{1_a, 2_a, \ldots, S_a\}$. The transition matrix Q_a is the identity matrix. For each state $i \in \{1 \ldots S\}$, we choose a best bandit a_i^* uniformly at random among the n bandits, independently for each i. The rewards of a state i_a are *i.i.d.* Bernoulli rewards with mean $\gamma(\tau, n)$ if $a \neq a_i^*$ and $\gamma'(\tau, n)$ if $a = a_i^*$. The initial distribution ρ couples the initial states of all bandits: for all $i \in \{1 \ldots S\}$,

$$\mathbb{P}\left(\forall a \in [n] : x_{0,a} = i_a\right) = \frac{1}{S}.$$

In this case, the Markovian bandit problem becomes a combination of S independent stochastic bandit problems with n bandits each. We denote by $\mathcal{M}_i^{\text{stoc.pb}}$ the random stochastic bandit problem for the initial state $\boldsymbol{i} = (i_a)_{a\in[n]}$. As the a_i^* are chosen independently, a learning algorithm $\mathcal{L}$ cannot use the information for $\mathcal{M}_i^{\text{stoc.pb}}$ to perform better on $\mathcal{M}_j^{\text{stoc.pb}}$, $j \neq i$.

Let ϕ be the distribution of the random Markovian bandit model $\mathcal{M}$ defined above and let T_i be the number of time steps spent in state $\boldsymbol{i}$ by the learning algorithm $\mathcal{L}$.

$$\begin{aligned}
\text{BayReg}(K, \mathcal{L}, \phi) &\geq \sum_{i=1}^{S} \mathbb{E}\left[\text{Reg}^{\text{stoc.pb}}(T_i, \mathcal{L}_i^{\text{stoc.pb}}, \mathcal{M}_i^{\text{stoc.pb}})\right] \\
&\geq \sum_{i=1}^{S} \mathbb{E}\left[\text{Reg}^{\text{stoc.pb}}(\tau, \mathcal{L}_i^{\text{stoc.pb}}, \mathcal{M}_i^{\text{stoc.pb}})\mathbb{I}_{\{T_i\geq\tau\}}\right] &(33)\\
&\geq \frac{S}{20}\sqrt{n\tau}\mathbb{P}\left(T_i \geq \tau\right) &(34)\\
&= \frac{1}{20\sqrt{2}}\mathbb{P}\left(T_i \geq \tau\right)\sqrt{\frac{SnK}{1-\beta}}, &(35)
\end{aligned}$$

where (33) is true because the expected regret is non-decreasing function of the number of episodes, (34) comes from (32) and (35) from the definition of τ.

We show in the Lemma 6 below that $\mathbb{P}\left(T_i \leq K/(2S(1-\beta))\right) \leq 8S/K$. This shows that for $K \geq 16S$, one has $\mathbb{P}\left(T_i \geq \tau\right) \geq 1/2$. This concludes the proof as $40\sqrt{2} \leq 60$.

Lemma 6. *Recall that T_i is the number of time steps that the MDP is in state $\boldsymbol{i}$ for the MDP model above. Let G_k be a sequence of* i.i.d. *Bernoulli random variable of mean $1/S$ and let H_k be an independent* i.i.d. *sequence of geometric random variable of parameter $1-\beta$. Then:*

(i) $T_i \sim \sum_{k=1}^{K} G_k H_k$,

(ii) $\mathbb{E}\left[T_i\right] = K/(S(1-\beta))$,

(iii) $\mathbb{P}\left(T_i \geq \mathbb{E}\left[T_i\right]/2\right) \geq 1 - 8S/K$.

Proof. Let G_k be a random variable that equals 1 if the initial state $\boldsymbol{i}$ is chosen at the beginning of episode k and recall that H_k is the episode length. By definition, the variables G_k and H_k are independent and follow respectively Bernoulli and geometric distribution. This shows *(i)*.

Let $W_k = G_k H_k$. As the W_k are *i.i.d.* and G_k and H_k are independent, we have:

$$\mathbb{E}\left[T_i\right] = K\mathbb{E}\left[H_1 G_1\right] = \frac{K}{S(1-\beta)}.$$

This shows (ii).

Moreover, $\mathrm{var}\left[T_i\right] = K\mathrm{var}\left[H_1 G_1\right]$. Hence, by using Chebyshev's inequality, one has:

$$\begin{aligned}
\mathbb{P}\left(T_i \leq \frac{\mathbb{E}\left[T_i\right]}{2}\right) &\leq \mathbb{P}\left(\left|T_i - \mathbb{E}\left[T_i\right]\right| \geq \frac{\mathbb{E}\left[T_i\right]}{2}\right) \\
&\leq \frac{4\mathrm{var}\left[T_i\right]}{\left(\mathbb{E}\left[T_i\right]\right)^2} \\
&= \frac{4}{K}\frac{\mathrm{var}\left[H_1 G_1\right]}{\left(\mathbb{E}\left[H_1 G_1\right]\right)^2}.
\end{aligned}$$

Concerning the variance, the second moment of a geometric random variable of parameter $1-\beta$ is $(1+\beta)/(1-\beta)^2$. This shows that $\mathbb{E}\left[(H_1 G_1)^2\right] = (1+\beta)/(S(1-\beta)^2) \leq 2S(\mathbb{E}\left[H_1 G_1\right])^2$. This implies:

$$\mathrm{var}\left[H_1 G_1\right] \leq (2S-1)(\mathbb{E}\left[H_1 G_1\right])^2 \leq 2S(\mathbb{E}\left[H_1 G_1\right])^2.$$

This implies (iii). □

C Proof of Theorem 2

Recall that π_* is the optimal policy of the unknown MDP $\mathcal{M}$ and that π_k is the policy that is used in episode k. π_k is optimal for the optimistic MDP that is chosen from the plausible MDP set $\mathbb{M}_k$:

$$\pi_k \in \arg\max_{\pi} \max_{\bar{\mathcal{M}} \in \mathbb{M}_k} V_{\bar{\mathcal{M}}}^{\pi}.$$

As [2], we argue that there exists a MDP $\bar{\mathcal{M}}_k \in \mathbb{M}_k$ such that π_k is an optimal policy for $\bar{\mathcal{M}}_k$. The definition of expected regret of MB-UCRL2 is

$$\begin{aligned}
\mathrm{Reg}(K, \text{MB-UCRL2}, \mathcal{M}) &= \sum_{k=1}^{K} \mathbb{E}\left[V_{\mathcal{M}}^{\pi_*}(\boldsymbol{X}_{t_k}) - V_{\mathcal{M}}^{\pi_k}(\boldsymbol{X}_{t_k})\right] \\
&= \sum_{k=1}^{K} \Big(\mathbb{E}\left[V_{\mathcal{M}}^{\pi_*}(\boldsymbol{X}_{t_k}) - V_{\bar{\mathcal{M}}_k}^{\pi_k}(\boldsymbol{X}_{t_k})\right] + \mathbb{E}\left[V_{\bar{\mathcal{M}}_k}^{\pi_k}(\boldsymbol{X}_{t_k}) - V_{\mathcal{M}}^{\pi_k}(\boldsymbol{X}_{t_k})\right]\Big),
\end{aligned} \tag{36}$$

where the expectation is taken over the random evolution of $\boldsymbol{X}_t$ when running the algorithm which determines the choice of $\bar{\mathcal{M}}_k$ and of π_k. For $0 < \delta < 1$, let $C'_k = \sqrt{2\log(4Snt_k/\delta)}$. For each episode k, the plausible MDP set $\mathbb{M}_k$ is defined by

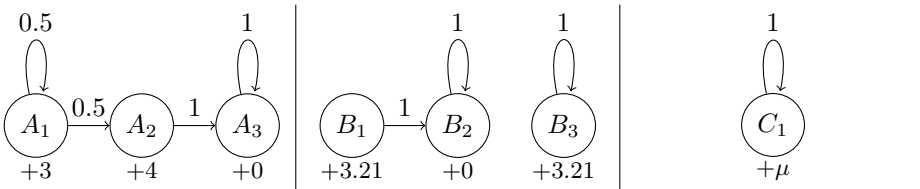
$$\mathbb{M}_k = \Bigg\{ (\bar{r}, \bar{Q}) : \forall a, x_a, |\bar{r}(x_a) - \hat{r}_k(x_a)| \leq \frac{C'_k}{\sqrt{4\max(1, N_k(x_a))}}, \text{ and } \Big\| \bar{Q}(x_a, \cdot) - \hat{Q}_k(x_a, .) \Big\|_1 \leq \frac{C'_k + 1.5\sqrt{S}}{\sqrt{\max(1, N_k(x_a))}} \Bigg\}. \tag{37}$$

Having (21) and (23), the unknown MDP $\mathcal{M}$ belongs to $\mathbb{M}_k$ for all $k \leq K$ with probability at least $1-\delta$. Since the extended value iteration preserves the OFU principle, we have $V^{\pi_k}_{\bar{\mathcal{M}}_k}(\rho) \geq V^{\pi^*}_{\mathcal{M}}(\rho)$ for each k if $\mathcal{M} \in \mathbb{M}_k$, which means that the first term of (36) is non-positive in this case. Consequently, the expected regret is bounded by the second term of (36) which is itself bounded by the distance between $\mathcal{M}$ and $\bar{\mathcal{M}}_k$ along the trajectory taken by the policy π_k as given in Equation (12). Combining the fact that $\mathcal{M} \in \mathbb{M}_k$ with probability at least $1-\delta$ with (12) and (36), we get

$$\mathrm{Reg}(K, \text{MB-UCRL2}, \mathcal{M}) \leq \mathbb{E}\left[\sum_{k=1}^{K} \sum_{t=t_k}^{t_{k+1}-1} \frac{C'_k + (2C'_k + 3\sqrt{S})H_k}{\sqrt{\max(1, N_k(X_{t,A_t}))}} \right], \tag{38}$$

with probability at least $1-\delta$. The detail analysis of the right term of (38) is the same as provided in Appendix A.5, which concludes the proof. Note that in case $\mathcal{M} \notin \mathbb{M}_k$, we can take $\delta = 1/K$ and bound the regret by $\frac{1}{(1-\beta)}$ as shown by (26).

D Proof of Theorem 4



(a) $\hat{Q}_a$ and $\hat{r}_a = r_a$. (b) $\hat{Q}_b = Q_b$ and $\hat{r}_b = r_b$. (c) $\hat{Q}_c = Q_c$ and $\hat{r}_c = r_c$.

Figure 2: Counterexample for OFU indices: $\hat{\mathcal{B}}_a$, $\hat{\mathcal{B}}_b = \mathcal{B}_b$, $\hat{\mathcal{B}}_c = \mathcal{B}_c$.

In this proof, we reason by contradiction and assume that there exists a procedure that computes local indices such that the obtained policy is such that for any estimate $\hat{\mathcal{B}}$ and any initial condition ρ, then if $\mathcal{M} \in \mathbb{M}(\hat{\mathcal{B}})$, one has

$$\sup_{\bar{\mathcal{M}} \in \mathbb{M}(\hat{\mathcal{B}})} V^{\pi_{I^{\hat{\mathcal{B}}}}}_{\bar{\mathcal{M}}}(\rho) \geq \sup_{\pi} V^{\pi}_{\mathcal{M}}(\rho). \tag{39}$$

In the remaining of this section, we set the discount factor to $\beta = 0.5$. For a given state x_a, we denote by $I(x_a)$ the local index of state x_a computed by this hypothetically optimal algorithm.

We first consider a Markovian bandit problem with two bandits $\{b, c\}$. We consider that these two bandits are perfectly estimated (*i.e.*, $\epsilon^r_b(x_b) = \epsilon^Q_b(x_b) = \epsilon^r_c(x_c) = \epsilon^Q_c(x_c) = 0$). The Markov chains for these bandits are depicted in Figure 2. Their transitions matrices and rewards are

$$Q_b = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } r_b = [3.21, 0, 3.21]; \qquad Q_c = [1] \text{ and } r_c = [\mu].$$

As the Markovian bandit are perfectly known, the indices $I(B_1)$, $I(B_2)$, $I(B_3)$ and $I(C_1)$ must be such that the obtained priority policy is optimal for the true MDP, that is: states B_1 and B_3 should have priority over C_1 (*i.e.*, $I(B_1) > I(C_1)$ and $I(B_3) > I(C_1)$) if and only if $\mu < 3.21$, and state

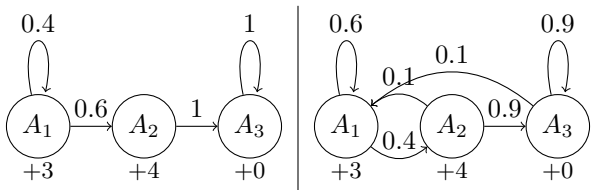
B_2 should have priority over C_1 (*i.e.*, $I(B_2) > I(C_1)$) if and only if $\mu < 0$. This implies that the local indices defined by our hypothetically optimal algorithm must satisfy

$$I(B_1) = I(B_3) > I(B_2).$$

Now, we consider Markovian bandit problems with two bandits $\{a, b\}$, where Bandit b is as before. For Bandit a, we consider a confidence set $\hat{\mathcal{B}}_a = (\hat{Q}_a, \hat{r}_a, \epsilon_a^r, \epsilon_a^Q)$ where $(\hat{Q}_a, \hat{r}_a)$ are depicted in Figure 2(a) and where $\epsilon_a^r(x_a) = 0$ and $\epsilon_a^Q(x_a) = 0.2$:

$$\hat{Q}_a = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \hat{\boldsymbol{r}}_a = \boldsymbol{r}_a = [3, 4, 0] \qquad \epsilon_a^Q = [0.1, 0.1, 0.1] \text{ and } \epsilon_a^r = [0, 0, 0].$$

We consider two possible instances of the "true" Markovian bandit problem, denoted $\mathcal{M}^1$ and $\mathcal{M}^2$. For $\mathcal{M}^1$, the transition matrix and reward function of the first bandit are depicted in Figure 3(a). For $\mathcal{M}^2$, they are depicted in Figure 3(b). In both cases, (Q_b, r_b) are as in Figure 2(b). It should be clear that $\mathcal{M}^1 \in \mathbb{M}$ and $\mathcal{M}^2 \in \mathbb{M}$.



(a) (Q^a, r^a) for $\mathcal{M}^1$ (b) (Q^a, r^a) for $\mathcal{M}^2$

Figure 3: The two instances of $\mathcal{B}_a^1$ and $\mathcal{B}_a^2$

If there exist indices that can be computed locally, then the indices for a bandit should not depend on the confidence that one has on the other bandits. The indices $I(A_1)$, $I(A_2)$ and $I(A_3)$ must satisfy the following facts:

- $I(A_3) \in (I(B_2), I(B_3))$ because for all Markovian bandit $\bar{\mathcal{M}} \in \mathbb{M}$, state A_3 should have priority over state B_2 and should not have priority over state B_3 (because of the discount factor $\beta = 1/2$).
- $I(A_2) > I(B_1) = I(B_3)$ because for all Markovian bandit $\bar{\mathcal{M}} \in \mathbb{M}$, state A_2 will give a higher instantaneous reward than state B_1 or B_3. It should therefore have a higher priority.

This leaves two possibilities for $I(A_1)$:

- If $I(A_1) > I(B_1) = I(B_3)$, then state A_1 has priority over both B_1 and B_3. We denote the corresponding priority policy π^1.
- If $I(A_1) < I(B_1) = I(B_3)$, then state B_1 and B_3 have a higher priority than state A_1. We denote the corresponding priority policy by π^2.

We use a numerical implementation of extended value iteration (available in the supplementary material) to find that:

$$\begin{aligned} \sup_{\bar{\mathcal{M}} \in \mathbb{M}} V_{\bar{\mathcal{M}}}^{\pi^2}(A_1, B_3) \approx 6.42 < \sup_{\pi} V_{\mathcal{M}^1}^{\pi}(A_1, B_3) \approx 6.47 \\ \sup_{\bar{\mathcal{M}} \in \mathbb{M}} V_{\bar{\mathcal{M}}}^{\pi^1}(A_1, B_1) \approx 5.96 < \sup_{\pi} V_{\mathcal{M}^2}^{\pi}(A_1, B_1) \approx 6.00 \end{aligned} \tag{40}$$

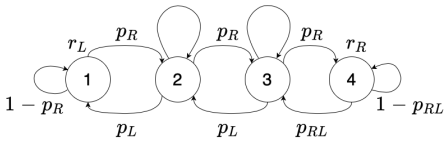
This implies that there does not exist any definition of indices such that (8) holds regardless of $\mathcal{M}$ and $\boldsymbol{x}$.

E Description of all Algorithms and Choice of Hyperparameter

In this section, we provide a detailed description of the simulation environment used in the paper. We first describe the Markov chain used in our example. Then, we describe all algorithms that we compare in the paper. For each algorithm, we give some details about our choice of hyperparameters. Last, we also describe the experimental methodology that we used in our simulations.

E.1 Description of the example

We design an environment with 3 bandits, all following a Markov chain represented in Table 1. This Markov chain is obtained by applying the optimal policy on the river swim MDP of [9]. This MDP is a classical example of an "hard-to-learn" MDP because it has reward only on edges. In each chain, there are 2 rewarding states: state 1 with low mean reward r_L, and state 4) with high mean reward r_R, both with Bernoulli distributions. At the beginning of each episode, all chains start in their state 1. Each chain is parametrized by the values of $p_L, p_R, p_{RL}, r_L, r_R$ that are given in Table 1 along with the corresponding Gittins indices of each chain.



					Gittins index for each state			
p_L	p_R	p_{RL}	r_L	r_R	1	2	3	4
0.1	0.2	0.3	0.2	1.0	0.276	0.2894	0.392	1.0
0.1	0.5	0.7	0.35	0.7	0.35	0.256	0.2892	0.7
0.1	0.4	0.5	0.4	0.65	0.4	0.250	0.286	0.65

Table 1: The random walk chain with 4 states. In state 4, the chain has an average reward r_R. For state 2 and 3, the chain gives zero reward. In state 1, the mean reward is r_L. This chain is obtained by applying the optimal policy on the 4-state river swim MDP of [9]. The table contains the parameters that we used, along with Gittins indices of all states when the discount factor is $\beta = 0.99$.

E.2 MB-PSRL

MB-PSRL, the adaption from PSRL, puts prior distribution on the parameters $(\boldsymbol{r}_a, Q_a)$ of each Bandit a, draws a sample from the posterior distribution and uses it to compute the Gittins indices at the start of each episode. We implement two posterior updates for the mean reward vector $\boldsymbol{r}_a$: Beta and Gaussian-Gamma. The second posterior, Gaussian-Gamma, will be used in prior choice sensitivity tests. For the transition matrix Q_a, we implemented Dirichlet posterior update because Dirichlet distribution is the only natural conjugate prior for categorical distribution. Beta, Gaussian-Gamma and Dirichlet distributions can be easily sampled using the numpy package of Python. This greatly contributes to the computational efficiency of MB-PSRL.

We give more details on this prior distribution and their conjugate posterior in the subsections below.

E.2.1 Bayesian Updates: Conjugate Prior and Posterior Distributions

MB-PSRL is a Bayesian learning algorithm. As such, it samples reward vectors and transition matrices at the start each episode. We would like to emphasize that neither the definition of the algorithm nor its performance guarantees that we prove in Theorem 1 depend on a specific form of the prior distribution ϕ. Yet, in practice, some prior distributions are more preferable because their conjugate distributions are easy to implement. In the following, we give concrete examples on how to update the conjugate distribution given the observations.

For $a \in [n]$ and $x_a \in \mathcal{S}_a$, let $N_k(x_a)$ be the number of activations of bandit a while in state x_a up to episode k. For this state x_a, the number of samples of the reward and of transitions from x_a are equal to $N_k(x_a)$. To ease the exposition, we drop the label a and assume that we are given:

- $N_k(x)$ *i.i.d.* samples $\{Y_1, \ldots, Y_{N_k(x)}\}$ of next states to which the bandit transitioned from x.
- $N_k(x)$ *i.i.d.* samples $\{R_1, \ldots, R_{N_k(x)}\}$ of random immediate rewards earned while the bandit was activated in state x

Each Y_i is such that $\mathbb{P}(Y_i = y) = Q(x, y)$ and each R_i is such that $\mathbb{E}[R_i] = r(x)$. In what follows, we describe natural priors that can be used to estimate the transition matrix and the reward vector.

E.2.2 Transition Matrix

If no information is known about the bandit, the natural prior distribution is to consider the lines $Q(x, \cdot)$ of the matrix as independent multivariate random variables uniformly distributed among all non-negative vectors of length S that sum to 1. This corresponds to a Dirichlet distribution of

parameters $\alpha = (1, \ldots, 1)$. For a given x, the variables $\{Y_1, \ldots, Y_{N_k(x)}\}$ are generated according to a categorical distribution $Q(x, \cdot)$. The Dirichlet distribution is self-conjugate with respect to the likelihood of a categorical distribution. So, the posterior distribution $\phi(Q(x, \cdot)|Y_1, \ldots, Y_{N_k(x)})$ is a Dirichlet distribution with parameters $\mathbf{c} = (c_1 \ldots c_S)$ where $c_y = 1 + \sum_{i=1}^{N_k(x)} \mathbb{I}_{\{Y_i = y\}}$.

E.2.3 Reward Distribution

As for the reward vector, the choice of a good prior depends on the distribution of rewards. We consider two classical examples: Bernoulli and Gaussian.

Bernoulli distribution A classical case is to assume that the reward distribution of a state x is Bernoulli with mean value $r(x)$. A classical prior in this case is to consider that $\{r(x)\}_{\{x \in \mathcal{S}\}}$ are *i.i.d.* random variables following a uniform distribution whose support is $[0, 1]$. The posterior distribution of $r(x)$ at time t is the distribution of $r(x)$ conditional to the reward observations from state x gathered up to time t. The posterior distribution $\phi(r(x) \mid R_1, \ldots, R_{N_k(x)})$ is then a Beta distribution with parameters $(1 + \sum_{i=1}^{N_k(x)} \mathbb{I}_{\{R_i = 1\}}, 1 + \sum_{i=1}^{N_k(x)} \mathbb{I}_{\{R_i = 0\}})$. Recall that the Beta distribution is a special case of the Dirichlet distribution in the same way as the Bernoulli distribution is a special case of the Categorical distribution.

Gaussian distribution We now consider the case of Gaussian rewards and we assume that the immediate rewards earned in state x are *i.i.d.* Gaussian random variables of mean and variance $(r(x), \sigma^2(x))$. A natural prior for Gaussian rewards is to consider that $\{(r(x), \frac{1}{\sigma^2(x)})\}_{\{x \in \mathcal{S}\}}$ are *i.i.d.* bivariate random variables where the marginal distribution of each $\frac{1}{\sigma^2(x)}$ is a Gamma distribution (it is a natural belief since the empirical variance of Gaussian has a chi-square distribution which is a special case of Gamma distribution). Conditioned on $\frac{1}{\sigma^2(x)}$, $r(x)$ follows a Gaussian distribution of variance $\sigma^2(x)$. We say that $(r(x), \frac{1}{\sigma^2(x)})$ has a Gaussian-Gamma distribution, which is self-conjugate with respect to a Gaussian likelihood (*i.e.,* the likelihood of Gaussian rewards). So, given the reward observations, the marginal distribution of $\frac{1}{\sigma^2(x)}$ is still a Gamma distribution. $r(x)$ has Gaussian distribution conditioned on the reward observations and $\frac{1}{\sigma^2(x)}$. Indeed, let $\hat{r}(x) = \frac{1}{N_k(x)} \sum_{i=1}^{N_k(x)} R_i$ and $\hat{\sigma}^2(x) = \frac{1}{N_k(x)} \sum_{i=1}^{N_k(x)} \left(R_i - \hat{r}(x)\right)^2$ be the empirical mean and empirical variance of R_i. Then it can be shown that the posterior distribution of $\frac{1}{\sigma^2(x)}$ and $r(x)$ are:

$$\frac{1}{\sigma^2(x)} \mid R_1, \ldots, R_{N_k(x)} \sim \text{Gamma}\left(\frac{N_k(x)+1}{2}, \frac{1}{2} + \frac{N_k(x)\hat{\sigma}^2(x)}{2} + \frac{N_k(x)\hat{r}^2(x)}{2(N_k(x)+1)}\right)$$

$$r(x) \mid \frac{1}{\sigma^2(x)}, R_1, \ldots, R_{N_k(x)} \sim \mathcal{N}\left(\frac{N_k(x)\hat{r}(x)}{N_k(x)+1}, \frac{\sigma^2(x)}{N_k(x)+1}\right).$$

For more details about the analysis of conjugate prior and posterior presented above as well as more conjugate distributions, we refer the reader to [10, 21].

Notice that a reward that has a Gaussian distribution violates the property that all rewards are in $[0, 1]$. This could invalidate the bound on the regret of our algorithm proven in Theorem 1. Actually, it is possible to correct the proof to cover the Gaussian case by replacing the Hoeffding's inequality used in Lemma 3 by a similar inequality, also valid for sub-Gaussian random variables, see [33]. In the experimental section (see F.3), we also show that a bad choice for the prior distribution of the reward (assuming a Gaussian distribution while the rewards are actually Bernoulli) does not alter too much the performance of the learning algorithm.

E.3 MB-UCRL2

For MB-UCRL2, we use the description of the Algorithm 2. This algorithm uses the confidence bounds for $\hat{r}$ and $\hat{Q}$ as defined in Section 4. To solve (6), we use the extended value iteration algorithm from [2].

E.4 GittinsQlearning from [8]

Similarly to MB-PSRL, the algorithm developed in [8] is specifically designed for learning Gittins indices. This algorithms uses the idea that the Gittins index of a state x_a is equal to the value function at x_a in a modified MDP, called the *restart-in-x_a* MDP (see [7] for the description of this MDP). The algorithm of [8] then uses Q-learning to learn these indices (which is why we choose to name it GittinsQLearning). For exploration, GittinsQlearning uses Softmax or Boltzmann distribution on the estimated Gittins indices at each decision time. For each Bandit a and each state $x_a \in \mathcal{S}_a$, we chose the learning rate of Q-learning to be $\frac{C_1}{C_2+N_t(x_a)}$ for restart-in-x_a MDP. This value of the learning rate is in accordance with the convergence conditions of Q-learning algorithms [17, 32, 20]. The best decreasing temperature of the Boltzmann distribution that we experimented has an hyperbolic shape: $\frac{C_3}{1+C_4 t}$.

Note that C_1, C_2, C_3 and C_4 are hyperparameters that must be tuned according to the current scenario. The author of [8] does not explain if there is a universally good choice of hyperparameters. In our numerical experiments, we tried various solutions and, for each experiment, we kept the values that give the best performance. Note that for some values of the hyperparameters (not reported here), the performance of GittinsQlearning was very bad: in particular the choice of the temperature that provides a good tradeoff between exploitation of good arms and exploration of others is quite difficult. This is an important difference with MB-PSRL in which the only hyperparameter is the prior distribution and we show in Appendix F.3 that the choice of prior has almost no influence on the performance. In practice, simply using a uniform prior makes MB-PSRL perform very well.

E.5 Vanilla implementation of PSRL [24]

Vanilla PSRL puts the prior distribution on the parameters (r, P) of the unknown global MDP, draws a sample from the posterior distribution, and computes the optimal policy of the sample. Similar to MB-PSRL, the specialized version, the prior belief is chosen for each state-action pair $(\boldsymbol{x}, a)$ according to the reward distribution and next state distribution. Hence, we implement exactly the same posterior updates for vanilla PSRL as for MB-PSRL.

There are two main differences between vanilla PSRL and MB-PSRL. First, the vanilla version does not use *a priori* knowledge on the structure of the MDP and therefore uses a prior on the global problem (it does not *a priori* assume that activating a bandit a will not change the state of bandit b for all $b \neq a$). The prior of vanilla version is therefore more general than the prior used by MB-PSRL which also means that it contains less information. Second, vanilla PSRL uses a generic algorithm to solve the MDP while MB-PSRL computes Gittins indices. Generic algorithms such as value or policy iteration takes at least $O(nS^n)$ of computation time. On the contrary, computing Gittins policy only takes $O(nS^3)$.

E.6 Vanilla implementation of UCRL2 [2]

At the start of each episode, UCRL2 uses the collected observations to compute the empirical estimators $\hat{r}_k(\boldsymbol{x}, a)$ and $\hat{P}_k(\boldsymbol{x}, a, \boldsymbol{y})$ of the true parameters $r(\boldsymbol{x}, a)$ and $P(\boldsymbol{x}, a, \boldsymbol{y})$ of the unknown MDP for $a \in [n], \boldsymbol{x}, \boldsymbol{y} \in \mathcal{E}$. UCRL2 constructs confidence sets, that is, the sets of r and P such that for all $a, n, k \geq 1$:

$$|r(\boldsymbol{x}, a) - \hat{r}_k(\boldsymbol{x}, a)| \leq \sqrt{\frac{7 \log(2S^n n t_k/\delta)}{2 \max\{1, N_k(\boldsymbol{x}, a)\}}} \text{ and } \|P(\boldsymbol{x}, a) - \hat{P}_k(\boldsymbol{x}, a)\|_1 \leq \sqrt{\frac{14 S^n \log(2n t_k/\delta)}{\max\{1, N_k(\boldsymbol{x}, a)\}}}. \tag{41}$$

UCRL2 then uses the extended value iteration algorithm to compute a policy π_k as

$$\pi_k \in \arg\max_{\pi \in \Pi} \sup_{\mathcal{M} \text{ that satisfy (41)}} V^{\pi}_{\mathcal{M}}(\rho).$$

The policy π_k is called *optimistic* as it is optimal for the MDP $\mathcal{M}_k$ that provides the largest value function while satisfying (41).

E.7 Experimental Methodology

In our numerical experiment, we did 3 scenarios to evaluate the algorithms (scenario 2 and 3 are given in Appendix F). In each scenario, we choose the discount factor $\beta = 0.99$ (which is classical) and we compute the regret over $K = 3000$ episodes. The number of simulations varies over scenario depending on how the regret is computed. For each run, we draw a sequence of horizons $\{H_k\}_{k\in[3000]}$ from a geometric distribution of parameter 0.01 and we run all algorithms for this sequence of time-horizons to remove a source of noise in the comparisons.

For a given sequence of policies π_k, following Equation (4), the expected regret is $\sum_{k=1}^{K} \Delta_k$ where Δ_k is the expected regret over episode k, $\Delta_k = \mathbb{E}\left[V_{\mathcal{M}}^{\pi_*}(\rho) - V_{\mathcal{M}}^{\pi_k}(\rho)\right]$. For a given Markovian bandit problem, the value $V_{\mathcal{M}}^{\pi^*}(\rho)$ can be computed by using the retirement evaluation presented in Page 272 of [36]. It seems, however, that the same methodology is not applicable to compute the value function of an index policy that is not the Gittins policy. This means that while the policy π_k is easily computable, we do not know of an efficient algorithm to compute its value $V_{\mathcal{M}}^{\pi_k}(\rho)$. Hence, in our simulations, we will use two methods to compute the regret, depending on the problem size:

1. (Exact method) Let (r_π, P_π) be the reward vector and transition matrix under policy π (i.e. $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{E}, r_\pi(\boldsymbol{x}) = r(\boldsymbol{x}, \pi(\boldsymbol{x})), P_\pi(\boldsymbol{x}, \boldsymbol{y}) = P(\boldsymbol{x}, \pi(\boldsymbol{x}), \boldsymbol{y})$). Using the Bellman equation, the value function under policy π is computed by

$$V_{\mathcal{M}}^{\pi} = (\mathbf{1} - \beta P_\pi)^{-1} r_\pi. \tag{42}$$

   The matrix inversion can be done efficiently with the numpy package of Python. However, this takes $S^{2n} + 2S^n$ of memory storage. Hence, when the number of states and bandits are too large, the exact computation method cannot be performed.

2. (Monte Carlo method) In Scenario 2, the model has $n = 9$ bandits with $S = 11$ states each, which makes the exact method inapplicable. In this case, it is still possible to compute the optimal policy and to apply Gittins index based algorithms but computing their value is intractable. In such a case, to measure the performance, we do 240 simulations for each algorithm and try to approximate Δ_k by

$$\hat{\Delta}_k = \frac{1}{\#\text{replicas}} \sum_{j=1}^{\#\text{replicas}} \sum_{t=0}^{H_k^{(j)}-1} \left[r(X^{*,(j)}_{t,A_t^{*,(j)}}) - r(X^{(j)}_{t,A_t^{(j)}}) \right], \tag{43}$$

   where $H_k^{(j)}$ is the horizon of the kth episode of the jth simulation and $\{X^{*,(j)}_{t,A_t^{*,(j)}}\}$ and $\{X^{(j)}_{t,A_t^{(j)}}\}$ are the trajectories of the oracle and the agent respectively. The term oracle refers to the agent that knows the optimal policy.

Note that the expectation of (43) is equal to the value given in (42) but (43) has a high variance. Hence, when applicable (Scenario 1 and 3) we use Equation (42) to compute the expected regret.

F Additional Numerical Experiments

F.1 Scenario 1: Small Dimensional Example (Random Walk chain)

This scenario is explained in Appendix E.1 and the main numerical results are presented in Section 7. Here, we provide the result with error bars with respect to the random seed. The error bar size equals twice the standard deviation over 80 samples (each sample is a simulation with a given random seed and the random seeds are different for different simulations).

F.2 Scenario 2: Higher Dimensional Example (Task Scheduling)

We now study an example that is too large to directly apply PSRL, UCRL2, and MB-UCRL2 . Hence, here we only compare MB-PSRL and GittinsQlearning.

We implement the environment proposed on page 19 of [8] that was used as a benchmark for GittinsQlearning in the cited paper. Each chain represents a task that needs to be executed, and is

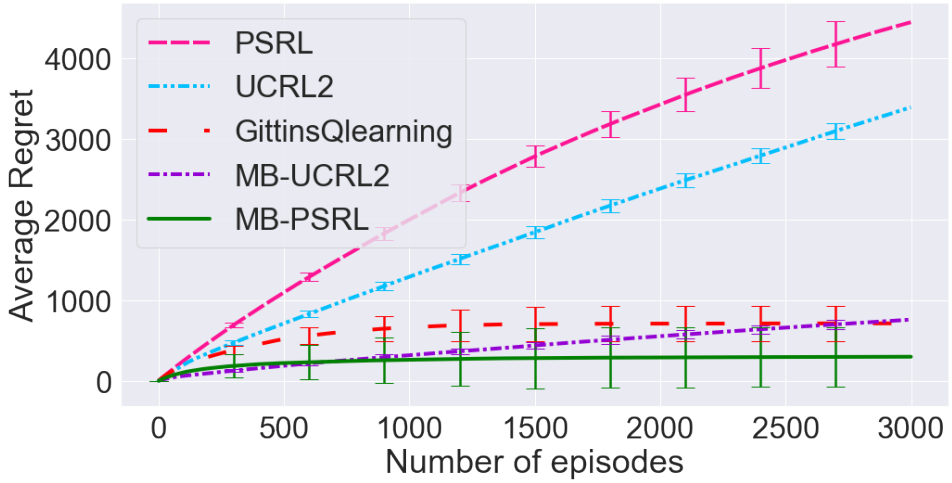
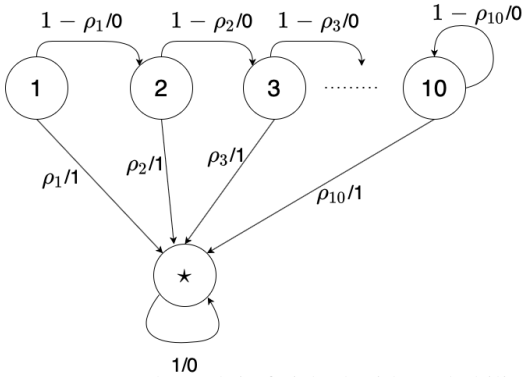


Figure 4: Average cumulative regret in function of the number of episodes. Result from 80 simulations in a Markovian bandit problem with three 4-state random walk chains given in Table 1. The horizontal axis is the number of episodes. The size of the error bar equals twice the standard deviation over 80 simulations.

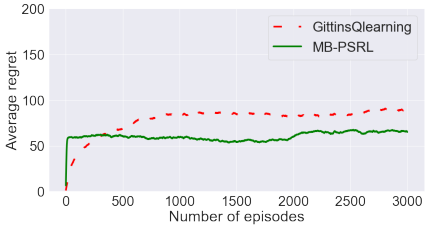
represented in Figure 5(a). Each task has 11 states (including finished state $\star$ that is absorbing). For a given chain $a \in \{1, \ldots, 9\}$ and a state $i \in \{1, \ldots, 10\}$, the probability that a task a ends at state i is $\rho_i^{(a)} = \mathbb{P}(\tau^{(a)} = i \mid \tau^{(a)} \geq i)$ where $\tau^{(a)}$ is the execution time of task a. We choose the same values of the parameters as in [8]: $\rho_1^{(a)} = 0.1a$ for $a \in \{1, \ldots, 9\}$, $\lambda = 0.8$, $\beta = 0.99$ and for $i \geq 2$,

$$\mathbb{P}\{x_a = i\} = \{1 - [1 - \rho_1^{(a)}]\lambda^{i-1}\}[1 - \rho_1^{(a)}]^{i-1}\lambda^{\frac{(i-1)(i-2)}{2}}.$$

Hence, the hazard rate $\rho_i^{(a)}$ is increasing with i. The reward in this scenario is deterministic: the agent receives 1 if the task is finished (*i.e.,* under the transition from any state i to state $\star$) and 0 otherwise (*i.e.,* any other transitions including the one from state $\star$ to itself). For MB-PSRL, we use a uniform prior for the expected rewards and consider that the rewards are Bernoulli distributed.



(a) In state i, the task is finished with probability ρ_i or transitions to state $i+1$ with probability $1-\rho_i$. For $i = 1, \ldots, 10$, the transition from state i to state $\star$ provides 1 as the immediate reward. Otherwise, the agent always receives 0 reward.



(b) Average cumulative regret over 240 simulations.

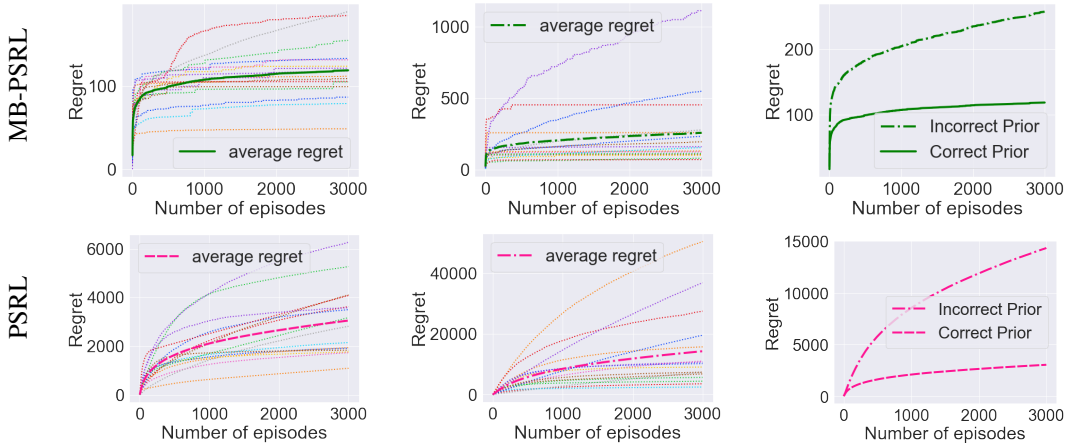
Figure 5: Task Scheduling with 11 states including the absorbing state (finished state).

The average regret of the two algorithms is displayed in Figure 5(b). As before, MB-PSRL outperforms GittinsQlearning by a small margin. Note that we also studied the time to run one simulation for 3000 episodes. This time is around 1 min for MB-PSRL and GittinsQlearning.

F.3 Scenario 3: Bayesian Regret and Sensitivity to the Prior

In this section, we study how robust the two implementations of PSRL are, namely MB-PSRL and vanilla PSRL (to simplify, we will just call the later PSRL), to a choice of prior distributions. As explained in Appendix E.2.3, the natural conjugate prior for Bernoulli reward is the Beta distribution. In this section, we simulate MB-PSRL and PSRL in which the rewards are Bernoulli but the conjugate prior used for the rewards are Gaussian-Gamma which is incorrect for Bernoulli random reward. In other words, MB-PSRL and PSRL have Gaussian-Gamma prior belief while the real rewards are Bernoulli random variables.

To conduct our experiments, we use a Markovian bandit problem with three 4-state random walk chains represented in Table 1. We draw 16 models by generating 16 pairs of (r_L, r_R) from $U[0, 1]$, 16 pairs of (p_L, p_R) from Dirichlet(3,(1,1,1)) and 16 values of p_{RL} from Dirichlet(2, (1,1)) for each chain. Each model is an unknown MDP that will be learned by MB-PSRL or PSRL. For each of these 16 models, we simulate MB-PSRL and PSRL 5 times with correct priors and 5 times with incorrect priors. The result can be found in Figure 6 which suggests that MB-PSRL performs better when the prior is correct and is relatively robust to the choice of priors in term of Bayesian regret. This figure also shows that PSRL seems more sensitive to the choice of prior distribution. Also note that for both MB-PSRL and PSRL, some trajectories deviate a lot from the mean, under correct priors but even more so with incorrect priors. This illustrates the general fact that learning can go wrong, but with a small probability.



(a) Correct Prior (b) Incorrect Prior (c) Bayesian Regret

Figure 6: Bayesian regret of MB-PSRL and vanilla PSRL in 3 4-state Random Walk chains. For each chain, we draw 16 random models and run the algorithms for 5 simulations in each model (there are 80 simulations in total). In panels (a) and (b), we plot 16 dotted lines that correspond to the average cumulative regret over 5 simulations in the 16 samples. The solid and dash-dot lines are the average regret each over 80 simulations (the estimated Bayesian regret). Figure 6a shows the performance when reward prior is well chosen (namely, $U([0, 1])$). Figure 6b is when the reward prior is incorrectly chosen (namely Gaussian-Gamma distribution). Figure 6c compares the Bayesian regret of the correct prior with the incorrect one (dash-dot line). In both case, the prior of next state transition is well chosen (namely, Dirichlet distribution). Y-axis range changes for each figure.

G Experimental environment

The code of all experiments is given in a separated zip file that contains all necessary material to reproduce the simulations and the figures.

Our experiments were run on HPC platform with 1 node of 16 cores of Xeon E5. The experiments were made using Python 3 and Nix and submitted as supplementary material and will be made publicly available with the full release of the paper. The package requirement are detailed in README.md. Using only *1 core* of Xeon E5, the Table 2 gives some orders of duration taken by each experiment

(with discount factor $\beta = 0.99$, and 3000 episodes per simulation). We would like to draw two remarks. First, the duration reported in Figure 1b is the time for policy computation (algorithm's parameters update and policy computation). The duration reported in Table 2 include this plus the computation time for oracle (because we track the regret), the state transition time along the trajectories of oracle and of each algorithm, resetting time... This explains why the duration reported in Table 2 cannot be compared to the duration reported in Figure 1b. Second, the duration shown in Table 2 are meant to be a rough estimation of the computation time (we only ran the simulation once and the average duration might fluctuate).

Experiment	MB-PSRL	GittinsQlearning	PSRL	MB-UCRL2	UCRL2	Total
Scenario 1	40 min	60 min	100 min	3days	19 days	22 days
Scenario 2	200 min	200 min	-	-	-	400 min
Scenario 3	90 min	-	260 min	-	-	350 min

Table 2: Approximative execution time for simulating each algorithm and tracking its regret in each scenario. This time includes the time given in Figure 1b and the computation time needed by oracle (because we track the regret), the state transition time along the trajectories of oracle and each algorithm, etc. In each scenario, we set the discount factor $\beta = 0.99$ and run the algorithms for 3000 episodes per simulation.