Online Markov Decision Processes Configuration with Continuous Decision Space

Davide Maran[*], Pierriccardo Olivieri[*], Francesco Emanuele Stradi[*],
Giuseppe Urso, Nicola Gatti, Marcello Restelli

Politecnico di Milano,
{davide.maran, pierriccardo.olivieri, francescoemanuele.stradi, nicola.gatti, marcello.restelli}@polimi.it,
giuseppe.urso@mail.polimi.it

Abstract

In this paper, we investigate the optimal online configuration of episodic Markov decision processes when the space of the possible configurations is continuous. Specifically, we study the interaction between a learner (referred to as the configurator) and an agent with a fixed, unknown policy, when the learner aims to minimize her losses by choosing transition functions in online fashion. The losses may be unrelated to the agent's rewards. This problem applies to many real-world scenarios where the learner seeks to manipulate the Markov decision process to her advantage. We study both deterministic and stochastic settings, where the losses are either fixed or sampled from an unknown probability distribution. We design two algorithms whose peculiarity is to rely on occupancy measures to explore with optimism the continuous space of transition functions, achieving constant regret in deterministic settings and $\sqrt{T}$ regret in stochastic settings, respectively. Moreover, we prove that the regret bound is tight with respect to any constant factor in deterministic settings. Finally, we compare the empiric performance of our algorithms with a baseline in synthetic experiments.

Introduction

Reinforcement Learning (RL) investigates the sequential interaction between a learner and an environment, aiming at continually improving the learner's strategy (Sutton and Barto 2018). In this context, the environment is customarily represented as a Markov Decision Process (MDP) with a fixed but unknown transition function. We study a general scenario where the interaction occurs in episodes, each with a predetermined length. Differently from the standard RL setting, we consider the learner not to be the agent playing the MDP, but the *configurator*. Precisely, at each episode, the learner picks the transition functions for the entire MDP (*i.e.*, a configuration) from a fixed continuous set. Next, she observes the loss suffered and the path traversed by the agent, which depend both on the agent's fixed policy and the transition chosen for the specific episode. The aim of the configurator is to minimize her regret between her total loss and that provided by an optimal fixed configuration.

Our model represents various real-world situations where the learner aims to manipulate the stochastic nature of the MDP to her advantage. For example, consider the sale of hotel rooms, where the MDP states are characterized by the number of rooms booked in different categories each day, while the transitions depend on the hotel's pricing and user behavior. Customarily, the hotels use a fixed pricing strategy that is trained offline and implemented online (technically, it is a post-training scenario). Given that users compare prices across hotels before booking rooms, a *competing* hotel (acting as the MDP configurator) can strategically adjust its pricing to influence user behavior and consequently alter the MDP transitions. Specifically, the competitor seeks to reduce the number of room reservations obtained by the agent to maximize her own.

Although this example illustrates an adversarial setting, our model applies to general scenarios that do not require a relationship between the configurator's loss and the agent's reward.

Related Work

Online learning in MDPs Several works initially introduced for on online learning (Cesa-Bianchi and Lugosi 2006; Hazan 2019) have been subsequently extended to MDPs (Auer, Jaksch, and Ortner 2008; Even-Dar, Kakade, and Mansour 2009; Neu et al. 2010). In particular, (Azar, Osband, and Munos 2017) study the problem of optimal exploration in episodic MDPs with unknown transitions and stochastic losses when the feedback is bandit. (Rosenberg and Mansour 2019a) study the online learning problem in episodic MDPs with adversarial losses and unknown transitions when the feedback is full information, presenting an online algorithm which provides a regret upper bound of $\tilde{\mathcal{O}}(\sqrt{T})$, where T is the number of episodes. (Rosenberg and Mansour 2019b) study the same setting when the feedback is bandit, providing a regret upper bound of $\tilde{\mathcal{O}}(T^{3/4})$, which was subsequently improved to $\tilde{\mathcal{O}}(\sqrt{T})$ by (Jin et al. 2019).

Configurable MDPs In MDPs, the transition function is customarily assumed to be fixed, see, *e.g.*, (Sutton and Barto 2018). However, various subsequent works represent environments with non-fixed transition probabilities, as provided in the works by (Satia and Lave 1973), (White and Eldeib 1994), and (Bueno et al. 2017). Recently, the concept of Configurable Markov Decision Processes (Conf-MDPs) was formalized by (Metelli, Mutti, and Restelli 2018). In particu-

[*] These authors contributed equally.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

lar, the authors propose an algorithm capable of optimizing, at the same time, the environment configuration, namely, the transition function and the policy of the learning agent. This line of research has been further expanded upon by (Metelli, Ghelfi, and Restelli 2019) and (Metelli, Manneschi, and Restelli 2022). Moreover, (Ramponi et al. 2021) extend the Conf-MDP setting to an online learning framework. This scenario involves a configurator who chooses online a transition function from a discrete set and aims to maximize her own reward, which is independent from the agent's one.

Adversarial Attacks Several works deal with adversarial attacks in MDPs, see, *e.g.*, (Ilahi et al. 2021). In the *bounded state attacks* framework, the adversary can manipulate the current state of an MDP in order to force the learning agent to make suboptimal decisions, see, *e.g.*, (Pattanaik et al. 2017), (Korkmaz 2021), and (Wu et al. 2022). Instead, in the *action attacks* setting, the adversary is capable of modifying the agent's actions, see, *e.g.*, (Lee et al. 2019), (Lee et al. 2021) and (Tan et al. 2020). Finally, in the *model attacks* framework, the attack consists in a (bounded) perturbation of the transition function of the MDP performed by an adversary, see, *e.g.*, (Rakhsha et al. 2020).

Original Contribution

We investigate the problem of *online configuration with continuous decision space* in MDPs, where the rewards may be both *deterministic* or *stochastic*. Precisely, we study the problem of an online configurator which chooses at any round a transition function from a continuous decision space and receives a loss which depends on both the configuration chosen and the fixed policy of the agent she is interacting with. First, we show that our setting can be seen as an instance of the well-known *Lipschitz bandit* framework, as well as a generalization of many post-training *adversarial attacks* models. Then, we propose two algorithms, namely, O-DOSC (Online Deterministic Optimistic Configuration Search) for deterministic settings and O-SOSC (Online Stochastic Optimistic Configuration Search) for the stochastic ones. We prove that O-DOSC achieves constant regret, matching the lower bound that we provide for the deterministic setting. Then, we show that O-SOSC achieves a $\widetilde{\mathcal{O}}\left(\sqrt{T}\right)$ regret bound in stochastic settings. Finally, we empirically validate our results with synthetic simulations.

Problem Formulation

Online MDPs

We introduce *online episodic loop-free* MDPs $\mathcal{M} = (X, A, P, \mathcal{R})$ defined as follows.

- T is the number of episodes, with $t \in [T]$ denoting a specific episode.
- X and A are the finite state and action spaces, respectively. By the loop-free property, X is partitioned into H layers $X_0, \dots, X_H$ such that the first and the last layers are singletons, *i.e.*, $X_0 = \{x_0\}$ and $X_H = \{x_H\}$. We will refer to H as the horizon. Moreover, we denote as $h(x)$ the layer of a specific state x.
- $P : X \times A \to \Delta(X)$ is the transition function, where, for ease of notation, we denote by $P(x'|x, a)$ the probability of going from state $x \in X$ to $x' \in X$ by taking action $a \in A$. By the loop-free property, it holds that $P(x'|x, a) > 0$ only if $x' \in X_{h+1}$ and $x \in X_h$ for some $h \in [0 \,..\, H-1]$.
- $\mathcal{R}$ is the reward function, which can be *deterministic*, that is, $\mathcal{R} : X \times A \to [0, 1]$, or *stochastic*, namely a distribution over $[0, 1]$ for every (x, a). We refer to the reward of a specific state-action pair $x \in X, a \in A$ for a specific episode $t \in [T]$ as $r_t(x, a)$.

Remark 1. *Any episodic MDP with horizon H that is* not *loop-free can be cast into a loop-free one by suitably duplicating the state space H times,* i.e.*, a state x is mapped to a set of new states (x, h), where $h \in [0 \,..\, H]$.*

A *policy* $\pi : X \to \Delta(A)$ defines a probability distribution over actions at each state. For ease of notation, we denote by $\pi(\cdot|x)$ the probability distribution for a state $x \in X$, with $\pi(a|x)$ denoting the probability of action $a \in A$.

Continuous Configurable-MDPs

The framework we propose, called Continuous Configurable-MDPs, is characterized by:

- an *agent*, which is characterized by the (optimal) policy π^* of a fixed MDP $\mathcal{M}(X, A, \overline{P}, \mathcal{R})$. We assume, without loss of generality, that π^* is deterministic, since it is well known that MDPs always admit an optimal deterministic policy;
- a *configurator*, which knows X, A, $\overline{P}$, H, T and at every episode $t \in [T]$ can choose a configuration (*i.e.*, a transition function) P_t from a bounded set $\mathcal{I}$, in order to minimize her loss. Similarly to the reward function, the loss function can be *deterministic*, that is, $\ell : X \times A \to [0, 1]$, or *stochastic*, namely a distribution over (x, a), still bounded in $[0, 1]$. We refer to the loss of a specific state-action pair $x \in X, a \in A$ for a specific episode $t \in [T]$ as $\ell_t(x, a)$. In the *stochastic setting*, $\ell_t(x, a)$ is a sample from $\mathcal{L}(x, a)$, drawn independently form the past. The average of loss distribution for (x, a), which does not depend on t, is denoted as $\ell(x, a) = \mathbb{E}[\mathcal{L}(x, a)]$.

Customarily in the literature, it is assumed that the configurator's loss is directly tied to the agent's reward, namely $\ell_t(x, a) = r_t(x, a)$ for every state-action pair and for every episode. Instead, in our setting, the two functions can be independent.

In Algorithm 1, we report the interaction between the agent and the configurator in the online MDP.

Precisely, at the beginning of each episode t, the loss function is either *deterministically* chosen (although this term may be slightly abused in this context) or *stochastically* chosen (refer to Line 2). Subsequently, the configurator chooses a transition function P_t (as in Line 3), and the MDP is initialized in the state x_0 (as per Line 4). During the episode, the agent traverses all the layers based on her policy π^* (as described in Line 6) and the transition P_t (as per Line 7). Upon completion of the episode, the configurator observes the complete trajectory and losses (as stated in Line 9).

Algorithm 1: Agent-Configurator Interaction

```
for $t \in [T]$ do
    For every state-action pair, $\ell_t(x,a) = \ell(x,a)$ in the *deterministic setting* or $\ell_t(x,a) \sim \mathcal{L}(x,a)$ in the *stochastic* setting
    configurator chooses $P_t \in \mathcal{I}$
    state is initialized to $x_0$
    for $h = 0, \dots, H-1$ do
        agent plays $a_h \sim \pi^*(\cdot|x_h)$
        environment evolves to $x_{h+1} \sim P_t(\cdot|x_h, a_h)$
    end for
    configurator observes $\{x_h, a_h\}_{h=0}^{H-1}$ and suffers $\{\ell_t(x_h, a_h)\}_{h=0}^{H-1}$
end for
```

Occupancy Measures

We introduce the notion of *occupancy measure*, see (Rosenberg and Mansour 2019b). Given a transition function P and a policy π, the occupancy measure $d^{P,\pi} \in [0,1]^{|X\times A\times X|}$ induced by P and π is such that, for every $x \in X_h$, $a \in A$, and $x' \in X_{h+1}$ with $h \in [0 \,..\, H-1]$:

$$d^{P,\pi}(x,a,x') = \mathbb{P}[x_h = x, a_h = a, x_{h+1} = x' | P, \pi]. \quad (1)$$

Moreover, we also define:

$$d^{P,\pi}(x,a) = \sum_{x' \in X_{h+1}} d^{P,\pi}(x,a,x'), \quad (2)$$

$$d^{P,\pi}(x) = \sum_{a \in A} d^{P,\pi}(x,a). \quad (3)$$

Then, we can introduce the following lemma, which characterizes *valid* occupancy measures.

Lemma 1. *(Rosenberg and Mansour 2019a) For every $d \in [0,1]^{|X\times A\times X|}$, it holds that d is a valid occupancy measure of an episodic loop-free MDP if and only if, for every $h \in [0 \,..\, H-1]$, the following three conditions hold:*

1. $\sum_{x\in X_h}\sum_{a\in A}\sum_{x'\in X_{h+1}} d(x,a,x') = 1$
2. $\sum_{a\in A}\sum_{x'\in X_{h+1}} d(x,a,x') = \sum_{x'\in X_{h-1}}\sum_{a\in A} d(x',a,x)$, $\forall x \in X_h$
3. $P^d = P$

where P is the transition function of the MDP and P^d is the one induced by d (see Equation (4)*).*

Notice that any occupancy measure d induces a transition function P^d and a policy π^d as:

$$P^d(x'|x,a) = \frac{d(x,a,x')}{d(x,a)}, \qquad \pi^d(a|x) = \frac{d(x,a)}{d(x)}. \quad (4)$$

Performance Metric

In order to have a proper performance metric for our algorithms, we introduce the notion of objective function of an MDP (in terms of loss).

Definition 1 (Expected Loss). *The expected loss suffered by the configurator at episode t is defined as the expected value of the sum of the losses given the configuration chosen. Namely,*

$$J_t^\pi(P) := \mathbb{E}\left[\sum_{h=1}^{H} \ell_t(x_h, a_h) \Big| \pi, P\right].$$

Thus, we define the cumulative regret as follows.

Definition 2 (Cumulative Regret). *The cumulative regret is defined as*

$$R_T := \sum_{t=1}^{T} J_t^\pi(P_t) - J_t^\pi(P^*),$$

where $P^* := \arg\min_{P\in\mathcal{I}} J_t^\pi(P)$.

Following the formulation based on the occupancy measure, the cumulative regret can be written as $R_T := \sum_{t=1}^T \ell^\top d^{P_t,\pi^*} - \min_{P\in\mathcal{I}} \sum_{t=1}^T \ell^\top d^{P,\pi^*}$, or equivalently, $R_T := \sum_{t=1}^T \ell^\top d^{P_t,\pi^*} - \min_{d\in\Delta(\mathcal{I},\pi^*)} \sum_{t=1}^T \ell^\top d$, where $d^{P,\pi}$ is the occupancy measure vector defined on the tuple (x,a) given a transition function P and a policy π, $\Delta(\mathcal{I},\pi^*)$ is the space of occupancy measures built given the fixed policy π^* and the transition function space $\mathcal{I}$, and ℓ is defined as:

- in the *deterministic* setting, ℓ is the loss vector composed by the loss values associated to each tuple (x,a), namely $\ell(x,a)$,
- in the *stochastic* setting, ℓ, is the vector composed by the expected values of the loss distribution for every (x,a), namely, $\mathbb{E}[\mathcal{L}(x,a)]$.

Given the definition of this setting, we aim that the regret is sublinear in T, namely $R_T = o(T)$.

The optimization problem described above is linear in the space of the occupancy measures, suggesting the potential adoption of online convex programming tools such as, *e.g.*, Bandit Linear Optimization (BLO) algorithms proposed by (Abernethy, Hazan, and Rakhlin 2008). However, these methods cannot be adopted to our case. Indeed, without the knowledge of the agent's policy, the configurator cannot compute the exact occupancy measure corresponding to her transition and the agent's policy, thus precluding the design of online bandit linear optimization algorithms working on the occupancy measure space. In particular, the configurator can only choose a transition function P_t and the objective function is highly nonlinear in the space of the transition functions.

Generality of the Setting and Interpretation

Our model captures various settings. In the following, we provide two different interpretations. The first focuses on MDPs with adversarial attacks, while the second focuses on Lipshitz bandits.

Interpreting Our Model as an MDP with Adversarial Attacks

Selecting the proper configurator's decision space $\mathcal{I}$, several forms of *adversarial attacks* in MDPs can be described by our model. Since the agent's policy is assumed to be fixed, our model captures post-training attacks.

In the following, the *adversarial attacks* modeled by our setting are presented with the associated configurator's continuous decision space.

- *Bounded state attacks*. The adversary can modify the agent's state, substituting it with another state that is similar to the original one. This can be modeled by setting:

  $$\mathcal{I} = \{P : \forall x \in X, a \in A, \exists x' \in B(x), \\ P(\cdot|x,a) = \overline{P}(\cdot|x',a)\},$$

  where $B(x) = \{x' : d(x,x') < \varepsilon\}$ for some distance function $d(\cdot)$ and $\varepsilon > 0$.
- *Action attacks*. Differently from the state attack scenarios, the adversary can perturb the action of the agent. This kind of attacks can be modeled by setting:

  $$\mathcal{I} = \{P : \forall x \in X, a \in A, \exists a' \in B(a), \\ P(\cdot|x,a) = \overline{P}(\cdot|x,a')\},$$

  where the set $B(a)$ is defined as in the case of bounded state attacks.
- *Model attacks*. The adversary can change the transition probabilities and the amount of the change is upper bounded according to some metrics. In particular, we adopt the *total variation metric*, denoted with TV. Therefore, $\mathcal{I}$ can be defined as

  $$\mathcal{I} = \{P : \text{TV}(P, \overline{P}) < \varepsilon\},$$

  for some $\varepsilon > 0$, where $\text{TV}(P,P') := \sum_{x\in X, a\in A} \|P(\cdot|x,a) - P'(\cdot|x,a)\|_1$.

Interpreting Our Model as a Lipschitz Bandit

We can show that the optimization problem faced by the configurator can be seen as a *Lipschitz bandit*, namely, the objective function in the optimization problem is *Lipschitz* continuous. This result is crucial in order to have a proper baseline to compare our theoretical guarantees to. Indeed, standard multi-armed bandits techniques cannot be applied to our setting, since the decision space is continuous.

The Lipschitz continuity of the objective is already well-known for discounted MDPs (Munos and Szepesvári 2008); nevertheless, we report here the result for the case of finite horizon problems.

Theorem 2. *Let P, P' be two transition functions, and π an arbitrary Markovian policy. Then,*

$$|J^\pi(P) - J^\pi(P')| \le \frac{H^2}{2} TV(P,P'),$$

where $TV(P,P') := \sum_{x\in X, a\in A} \|P(\cdot|x,a) - P'(\cdot|x,a)\|_1$.

Theorem 2 suggests that algorithms for Lipschitz bandits can be used to solve our problem. In the specific case of the Zooming algorithm by (Kleinberg, Slivkins, and Upfal 2008)—one of the state-of-the-art algorithms for Lipschitz bandits—, we can derive the following upper regret bound.

Corollary 3. *The Zooming algorithm in our setting achieves a regret of $R_T \le T^{\frac{1+\mathcal{D}(\mathcal{I})}{2+\mathcal{D}(\mathcal{I})}}$, where $\mathcal{D}(\mathcal{I})$ is the Zooming dimension of the space $\mathcal{I}$.*

When the decision space $\mathcal{I}$ depends on a family of p continuous parameters, its Zooming dimension is exactly p, so that the regret becomes $T^{\frac{1+p}{2+p}}$. As we show in the following, this regret bound can be dramatically improved and therefore the Zooming algorithm is suboptimal for our problem.

Deterministic Settings

We focus on deterministic settings, and we present our algorithm and its theoretical guarantees. More precisely, we assume there is a fixed function $\ell : X \times A \to [0,1]$, such that the configurator will always achieve the same loss whenever the agent chooses a particular action in a given state.

Algorithm

Algorithm 2 provides the pseudo-code of *Online Deterministic Optimistic Configuration Search* (O-DOSC), which tackles deterministic losses. As is customary in the online learning, the configurator needs to face an exploration-exploitation trade-off when searching for the optimal configuration. Specifically, the choice of P_t needs to balance the exploration of unobserved states with the minimization of the configurator's losses.

As stated above, we assume that the optimal policy π^* in the MDP is deterministic. Thus, our algorithm can safely keep track of the actions played and losses obtained. For this purpose, the set Π is initialized to contain all possible deterministic policies, while the function $\widehat{\ell}$ is initialized to return a loss value of 0 for every tuple (x,a) (Lines 1–2). Such an initialization for the function $\widehat{\ell}$ is chosen to guarantee optimism vs. uncertainty with respect to the actual loss function.

In order to determine the transition function P_t for each episode, an optimistic approach is adopted. In particular, we minimize the objective over the space of the occupancy measures, which is based on an estimate of the agent's policy (as reported in Line 4). This approach is optimistic with respect to both the policy and the loss function, which is set to be 0 when non-visited. Additionally, it is possible to simplify the optimization over $\mathcal{I}$ and Π by reducing to the optimization over the space $\Delta(\mathcal{I},\Pi)$, where $d^{P,\pi} \in \Delta(\mathcal{I},\Pi)$. For a detailed study of the computational complexity of the minimization update, please refer to the Appendix.

Then, once the agent's trajectory and losses suffered throughout the path have been observed (Line 5), the following updates are performed. For $\widehat{\ell}$, the 0 values associated with the tuples (x,a) visited during the episode are substituted with the observed losses (Line 6). Instead, for the set Π, the actions of the state traversed but not executed by the agent are discarded from the set (Line 7).

Algorithm 2: O-DOSC Algorithm

```
Require: $X, A, H, \mathcal{I}$
$\Pi \leftarrow$ set of all deterministic policies
$\widehat{\ell}(x,a) \leftarrow 0 \; \forall (x,a) \in X \times A$
for $t \in [T]$ do
    Choose $P_t = \arg\min_{P \in \mathcal{I}, \pi \in \Pi} \sum_{x,a} \left( \sum_{x' \in X_{h(x)+1}} d^{P,\pi}(x,a,x') \right) \widehat{\ell}(x,a)$
    Observe $\{x_h, a_h, \ell(x_h, a_h)\}_{h=0}^{H-1}$
    $\widehat{\ell}(x_h, a_h) \leftarrow \ell(x_h, a_h) \quad \forall h \in [0..H-1]$
    $\Pi \leftarrow \Pi \setminus \{x_h, a\}_{\forall a \neq a_h, \forall h \in [0 \,..\, H-1]}$
end for
```

Upper and Lower Regret Bounds

In this section, we present the theoretical guarantees of our O-DOSC algorithm in deterministic settings. Initially, we state the regret bound achieved by our algorithm, and, subsequently, we show that the regret bound matches the lower bound for our specific setting.

In deterministic settings, we show that Algorithm 2 achieves a constant regret bound.

Theorem 4. *In deterministic settings, Algorithm 2 guarantees a regret upper bound*

$$R_T \leq (H+1)|X|.$$

Proof Sketch. First, notice that, due to the deterministic nature of the loss, visiting each state one time is sufficient to compute the optimal policy.

Still, it is necessary to deal with states for which the associated occupancy measure is very low under every transition $P \in \mathcal{I}$ available to the configurator. Since these states are unlikely to be visited in the whole episode, they may prevent the configurator from learning the optimal configuration.

The key observation is that the aforementioned states cannot contribute significantly to the regret: defining the estimation error as follows, $\varepsilon_t := J^{\pi}(P_t) - LB_t(P_t)$, where $LB_t(P_t) := \min_{P \in \mathcal{I}, \pi \in \Pi} \sum_x d^{P,\pi}(x) \widehat{\ell}(x)$, we can show that,

1. The regret is bounded by $R_T \leq \mathbb{E}\left[\sum_{t=1}^{T} \varepsilon_t\right]$
2. At each step, the undiscovered states $\overline{X}_t$ satisfy $\sum_{x \in \overline{X}_t} d^{P_t,\pi}(x) \geq \frac{\varepsilon_t}{H+1}$

Given the previous steps, it can be derived that,

$$R_T \leq (H+1) \sum_{x \in \overline{X}_t} d^{P_t,\pi}(x),$$

which implies that the higher regret at one step, the higher probability to discover more states, thus lower regret for the future. □

The previous result is rather intuitive. Indeed, since both the optimal policy and reward function are deterministic, once the configurator visited the entire MDP, the optimal configuration has been found.

The reader may wonder if the the regret bound shown in Theorem 4 is tight for the setting. In the following, we show that our result is the best *any* algorithm can achieve. Therefore, Algorithm 2 matches the lower bound of the deterministic setting. Indeed, we can show that,

Theorem 5. *In deterministic settings, any algorithm achieves a regret of order* $\Omega(H|X|)$.

Proof Sketch. We properly build a Markov decision process as follows: the second layer is composed by approximately $|X|$ states while the remaining layers by two states only. The agent policy chooses a non-trivial action only at the second step, after which the loss in uniquely determined. Thus, the following $H-3$ steps have the effect of increasing the loss $H-3$ times. In order to get to the optimal state (in the second layer), the configurator has to pull in expectation approximately $|X|/2$ suboptimal configurations, thus augmenting the regret of a $H-3$ factor. This in turns implies a regret of approximately $(H-3)|X|/2$. □

Stochastic Settings

We focus on stochastic settings, and we present our algorithm and its theoretical guarantees. Precisely, we assume that there is a fixed probability distribution for every state-action pair, denoted as $\mathcal{L}(x,a)$, which drives the sampling of losses from the interval $[0,1]$ every time the agent chooses an action in a given state.

Algorithm

Algorithm 3 provides the pseudo-code of *Online Stochastic Optimistic Configuration Search* (O-SOSC) for stochastic losses. Similarly to what happens in deterministic settings, the configurator needs to address an exploration-exploitation trade-off when seeking for the optimal configuration. Again, the choice of P_t is required to balance the exploration of non-visited states with the minimization of the configurator's losses. Furthermore, in this case we introduce an additional complexity, given by the way losses are chosen.

By the theory of MDPs, we can safely assume that the optimal policy π^* for the MDP is deterministic. Algorithm 3 keeps track of the action played and the losses obtained by the configurator. For this purpose, the set Π is initialized to containing all possible deterministic policies, while $\widehat{\ell}_t$ is initialized to return a loss value of 0 for every tuple (x,a) (Lines 1–2). We choose this initialization for the function $\widehat{\ell}_t$ to be optimistic with respect to the actual loss function.

To determine the transition function P_t for each episode, we take an optimistic approach by minimizing the objective over the space of occupancy measures based on an estimate of the agent's policy (as reported in Line 4). It is worth noting that this update is optimistic with respect to both the policy and the loss function, which is set to 0 when non-visited, and is computed with UCB-like lower bound once traversed. Moreover, it is possible to simplify the optimization over $\mathcal{I}$ and Π by reducing it to the optimization over the space $\Delta(\mathcal{I}, \Pi)$, where $d^{P,\pi} \in \Delta(\mathcal{I}, \Pi)$. For a detailed

Algorithm 3: O-SOSC Algorithm

```
Require: $X, A, H, \mathcal{I}, \delta, T$
1: $\Pi \leftarrow$ set of all deterministic policies
2: $\widehat{\ell}_1(x,a) \leftarrow 0 \quad \forall (x,a) \in X \times A$
3: for $t \in [T]$ do
4:     Choose $P_t = \arg\min_{P \in \mathcal{I}, \pi \in \Pi} \sum_{x,a} \left( \sum_{x' \in X_{h(x)+1}} d^{P,\pi}(x,a,x') \right) \widehat{\ell}_t(x,a)$
5:     Observe $\{x_h, a_h, \ell_t(x_h,a_h)\}_{h=0}^{H-1}$
6:     Bonus $= \sqrt{\frac{-\log(\delta)+\log(N_t(x_h)(N_t(x_h)+1))}{2N_t(x_h)}}$
7:     $\widehat{\ell}_{t+1}(x_h,a_h) \leftarrow \max\left(0, \overline{\ell}_t(x_h,a_h) - \text{Bonus}\right) \quad \forall h \in [0..H-1]$
8:     $\Pi \leftarrow \Pi \setminus \{x_h, a\}_{\forall a \neq a_h, \forall h \in [0\ ..\ H-1]}$
9: end for
```

study of the computational complexity of the minimization update, please refer to the Appendix.

Once the agent's trajectory and losses suffered throughout the path have been observed (Line 5), the following updates are performed. For $\widehat{\ell}_t$, the values associated with the tuples (x,a) visited during the episode are updated with a UCB-like term that depends on the number of visits of a specific state $N_t(x)$ (Line 6), which is subtracted to the empirical mean $\overline{\ell}(x,a)$ of the losses observed. For the set Π, the actions of the state traversed but not executed by the agent are discarded from the set (Line 7).

Theoretical Guarantees

In this section, we present the theoretical result for Algorithm 3. First of all, we can derive a simple lower bound of the regret in the stochastic case.

Theorem 6. *In stochastic settings, any algorithm achieves a regret of order* $\Omega(\sqrt{|X|T})$.

Proof. In order to find a proper lower bound, we restrict to the (simpler) discrete decision space case. In such a setting, our model can be seen as a generalization of the multi-armed bandit setting. Specifically, given any multi-armed bandit problem, we can build an equivalent instance of our problem as follows. For every arm of the bandit problem, we have a transition function in $\mathcal{I}$ bringing deterministically from a common initial state to a different state. This implies that the number of transition functions in the MDP equals the number of arms of the bandit problem ($|\mathcal{I}| = |X|$). Therefore, the standard instance independent lower bound for multi-armed bandits with $|\mathcal{I}| = |X|$ number of arms leads to a regret of $R_T = \Omega(\sqrt{|X|T})$ which represents a lower bound for our problem. □

Finally, we show that Algorithm 3 achieves a sublinear regret bound that matches the aforementioned lower bound in the number of episodes T.

Theorem 7. *In the stochastic setting, for the choice* $\delta = T^{-1/2}$*, Algorithm 3 achieves a regret upper bounded as follows,*

$$R_T = \widetilde{\mathcal{O}}\left(|X|\sqrt{T} + H|X|\right).$$

Proof Sketch. In the *stochastic* setting, the configurator has to visit each state more than one time, due to the stochastic nature of the loss.

Analogously to stochastic multi-armed bandits, the aforementioned issue is faced defining a *good event* under which the configurator is able to learn a good policy by only visiting all the states for a fixed and not too large amount of times.

We refer to the *good events* s E^c, where E corresponds to

$$E := \left\{ \exists x \in X, t \in [T] : \ |\overline{\ell}_t(x) - \ell(x)| > \sqrt{\frac{-\log(\delta) + \log(N_t(x)(N_t(x)+1))}{2N_t(x)}} \right\},$$

and where $N_t(x)$ is the number of times state x has been visited by round t, and $\delta = T^{-1/2}$. With a standard concentration inequality and a union bound, we can prove that $\mathbb{P}(E) \leq 2|X|\delta$. Next, it is necessary to focus on the *good event* E^c.

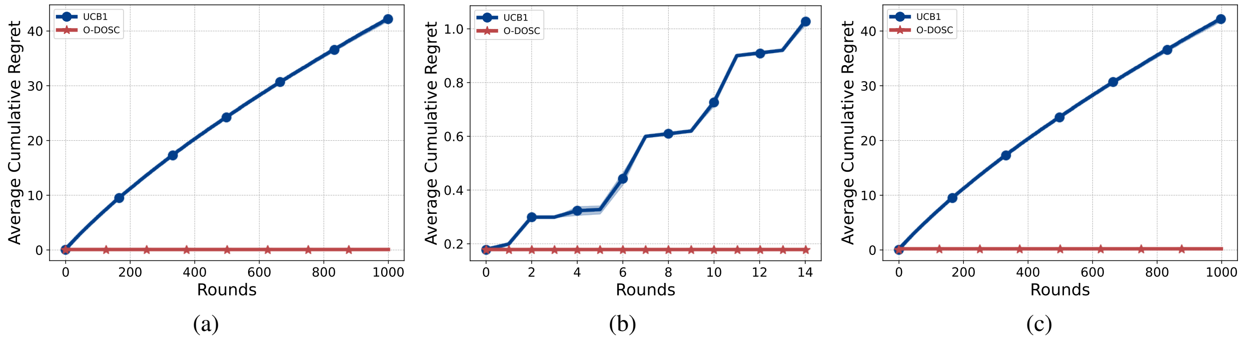
Under E^c, the proof follows the one for the deterministic case with a fixed number of visits for every state, showing that the expected regret conditioned to the event is of order $\widetilde{\mathcal{O}}\left(|X|\sqrt{T} + H|X|\right)$. □

Empirical Evaluation

In this section, we experimentally evaluate the performance of Algorithms 2 and 3 in terms of empiric regret. We describe the results obtained in the deterministic and stochastic settings separately. In each case, we conduct experiments with both discrete and continuous decision spaces $\mathcal{I}$.

As a baseline, we opt for UCB1 (Auer, Cesa-Bianchi, and Fischer 2002) since, in the case of discrete decision spaces, UCB1 is a standard baseline, while, in the case of continuous decision spaces, UCB1 can be preferred to Zooming (Kleinberg, Slivkins, and Upfal 2008) for two reasons. The first reason is that the design of a suitable covering oracle for Zooming raises several conceptual and computational issues due to the high number of dimensions whose solutions is open. The second reason is that, in our experimental settings, the optimal solution is one of the arms, and, in these cases, UCB1 is a more severe baseline than Zooming as it guarantees a much better regret bound.

In the following experiments, we consider a Markov decision process structured as follows. The MDP consists of four layers. As is standard in the loop-free model, the first and the last layers are singletons, while the second and third layers each comprise two states. Additionally, every state is associated with two actions. For reasons of space, the description of the experimental settings and additional details on the experimental results can be found in the Appendix.

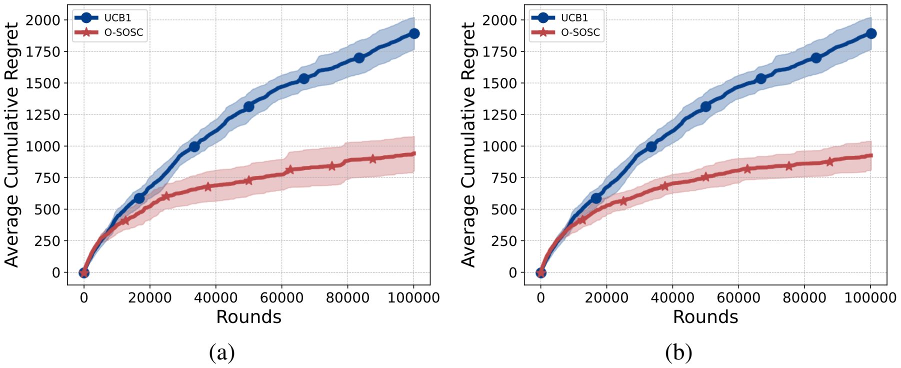


(a)

(b)

(c)

Figure 1: Average cumulative regret with a 95% confidence interval over 10 experiments in deterministic settings with discrete (a, b) and continuous (c) decision spaces.



(a)

(b)

Figure 2: Average cumulative regret with a 95% confidence interval over 10 experiments in stochastic settings with discrete (a) and continuous (b) decision spaces.

Deterministic Settings We report in Figure 1 the experimental results obtained with deterministic settings where the cumulative regret is averaged over 10 runs.

In particular, Figure 1(a) shows the results with discrete settings, while Figure 1(c) shows the results with continuous settings. In both cases, O-DOSC dramatically outperforms UCB1. Figure 1(b) clearly shows that O-DOSC effectively computes the optimal transition function during the very initial rounds, and subsequently it ceases to explore. Indeed, once O-DOSC visited all the states, it can numerically compute the optimal transition. Instead, UBC1 keeps exploring for a long time.

Stochastic Settings We report in Figure 2 the experimental results obtained with deterministic settings where the cumulative regret is averaged over 10 runs.

Precisely, Figure 2(a) shows the results with discrete settings, while Figure 2(b) shows the results with continuous settings.In both cases, O-SOSC outperforms UCB1. Differently from what happens in deterministic settings, O-SOSC does not find the optimal solution in the initial rounds, and additional exploration is required. However, the performance exhibited by O-SOSC in this setting is remarkably impressive.

Conclusions

In this paper, we propose the problem of *online configuration* of Markov decision processes with *continuous decision spaces*. We study the problem both when the losses are deterministic and stochastic. We propose O-DOSC algorithm, which achieves constant regret in deterministic settings, and we show that this result is tight for any constant with respect to the lower bound. Then, we propose O-SOSC which achieves a sublinear regret bound when the losses are stochastic. Finally, we empirically validate our theoretical results with synthetic simulations.

Future Works In future work, we are interested in studying the problem when losses are *adversarial*, namely no statistical assumption are made. Furthermore, we aim to study the problem of *online configurations* against a learning agent, namely, when the policy of the agent is allowed to be dynamic. In such a setting, we are interested in understanding how the results change when the agent is employing a no-regret optimizer and when she is omniscient, namely, she can observe the transitions chosen by the configurator and then commit to a policy as in a Stackelberg game.

References

Abernethy, J. D.; Hazan, E.; and Rakhlin, A. 2008. Competing in the Dark: An Efficient Algorithm for Bandit Linear Optimization. In *Annual Conference Computational Learning Theory*.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2): 235–256.

Auer, P.; Jaksch, T.; and Ortner, R. 2008. Near-optimal Regret Bounds for Reinforcement Learning. In Koller, D.; Schuurmans, D.; Bengio, Y.; and Bottou, L., eds., *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc.

Azar, M. G.; Osband, I.; and Munos, R. 2017. Minimax Regret Bounds for Reinforcement Learning.

Bueno, T. P.; Mauá, D. D.; Barros, L. N.; and Cozman, F. G. 2017. Modeling Markov Decision Processes with Imprecise Probabilities Using Probabilistic Logic Programming. In Antonucci, A.; Corani, G.; Couso, I.; and Destercke, S., eds., *Proceedings of the Tenth International Symposium on Imprecise Probability: Theories and Applications*, volume 62 of *Proceedings of Machine Learning Research*, 49–60. PMLR.

Cesa-Bianchi, N.; and Lugosi, G. 2006. *Prediction, learning, and games*. Cambridge university press.

Even-Dar, E.; Kakade, S. M.; and Mansour, Y. 2009. Online Markov decision processes. *Mathematics of Operations Research*, 34(3): 726–736.

Hazan, E. 2019. Introduction to Online Convex Optimization. *CoRR*, abs/1909.05207.

Ilahi, I.; Usama, M.; Qadir, J.; Janjua, M. U.; Al-Fuqaha, A.; Hoang, D. T.; and Niyato, D. 2021. Challenges and Countermeasures for Adversarial Attacks on Deep Reinforcement Learning. arXiv:2001.09684.

Jin, C.; Jin, T.; Luo, H.; Sra, S.; and Yu, T. 2019. Learning Adversarial MDPs with Bandit Feedback and Unknown Transition.

Kleinberg, R.; Slivkins, A.; and Upfal, E. 2008. Multi-Armed Bandits in Metric Spaces. arXiv:0809.4882.

Korkmaz, E. 2021. Investigating Vulnerabilities of Deep Neural Policies. arXiv:2108.13093.

Lee, X. Y.; Esfandiari, Y.; Tan, K. L.; and Sarkar, S. 2021. Query-based Targeted Action-Space Adversarial Policies on Deep Reinforcement Learning Agents. arXiv:2011.07114.

Lee, X. Y.; Ghadai, S.; Tan, K. L.; Hegde, C.; and Sarkar, S. 2019. Spatiotemporally Constrained Action Space Attacks on Deep Reinforcement Learning Agents. arXiv:1909.02583.

Metelli, A. M.; Ghelfi, E.; and Restelli, M. 2019. Reinforcement Learning in Configurable Continuous Environments. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 4546–4555. PMLR.

Metelli, A. M.; Manneschi, G.; and Restelli, M. 2022. Policy space identification in configurable environments. *Machine Learning*, 111(6): 2093–2145.

Metelli, A. M.; Mutti, M.; and Restelli, M. 2018. Configurable Markov Decision Processes. arXiv:1806.05415.

Munos, R.; and Szepesvári, C. 2008. Finite-Time Bounds for Fitted Value Iteration. *Journal of Machine Learning Research*, 9(5).

Neu, G.; Antos, A.; György, A.; and Szepesvári, C. 2010. Online Markov decision processes under bandit feedback. *Advances in Neural Information Processing Systems*, 23.

Pattanaik, A.; Tang, Z.; Liu, S.; Bommannan, G.; and Chowdhary, G. 2017. Robust Deep Reinforcement Learning with Adversarial Attacks. arXiv:1712.03632.

Rakhsha, A.; Radanovic, G.; Devidze, R.; Zhu, X.; and Singla, A. 2020. Policy Teaching via Environment Poisoning: Training-time Adversarial Attacks against Reinforcement Learning. arXiv:2003.12909.

Ramponi, G.; Metelli, A. M.; Concetti, A.; and Restelli, M. 2021. Learning in Non-Cooperative Configurable Markov Decision Processes. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 22808–22821. Curran Associates, Inc.

Rosenberg, A.; and Mansour, Y. 2019a. Online Convex Optimization in Adversarial Markov Decision Processes. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 5478–5486. PMLR.

Rosenberg, A.; and Mansour, Y. 2019b. Online Stochastic Shortest Path with Bandit Feedback and Unknown Transition Function. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Satia, J. K.; and Lave, R. E. 1973. Markovian Decision Processes with Uncertain Transition Probabilities. *Operations Research*, 21(3): 728–740.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book. ISBN 0262039249.

Tan, K. L.; Esfandiari, Y.; Lee, X. Y.; Aakanksha; and Sarkar, S. 2020. Robustifying Reinforcement Learning Agents via Action Space Adversarial Training. arXiv:2007.07176.

White, C. C.; and Eldeib, H. K. 1994. Markov Decision Processes with Imprecise Transition Probabilities. *Operations Research*, 42(4): 739–749.

Wu, F.; Li, L.; Huang, Z.; Vorobeychik, Y.; Zhao, D.; and Li, B. 2022. CROP: Certifying Robust Policies for Reinforcement Learning through Functional Smoothing. arXiv:2106.09292.

Computational Complexity of the Minimization Problem

In this section we study the computational complexity of the minimization update performed by Algorithm 2 and Algorithm 3. Indeed, the optimization problem required to be solved strongly depends on how the decision space of the transition function is chosen beforehand. In the following, we show that the minimization update can be performed in polynomial time when the decision space $\mathcal{I}$ is:

- $\mathcal{I} = \big\{P : |P(x'|x,a) - \overline{P}(x'|x,a)| \leq \epsilon(x,a,x'),\ \ \forall(x,a,x') \in X_h \times A \times X_{h+1}\big\}$
- $\mathcal{I} = \big\{P : ||P(\cdot|x,a) - \overline{P}(\cdot|x,a)||_1 \leq \epsilon(x,a),\ \ \forall(x,a) \in X \times A\big\}$
- $\mathcal{I}$ is a discrete set.

Thus, we show that the *O-DOSC* and *O-SOSC* optimization problems applied to the previous decision spaces may be modeled as Linear Programs (or a combination of them), which implies that they can be solved in polynomial time.

Precisely, the optimization problem that has to be solved in Algorithm 2 is the following:

$$\underset{P\in\mathcal{I},\pi\in\Pi}{\arg\min} \sum_{x,a}\left(\sum_{x'\in X_{h(x)+1}} d^{P,\pi}(x,a,x')\right)\widehat{\ell}(x,a)$$

The idea is to optimize on the occupancy space $\Delta(\mathcal{I},\Pi)$, namely:

$$\underset{d\in\Delta(\mathcal{I},\Pi)}{\arg\min} \sum_{x,a}\left(\sum_{x'\in X_{h(x)+1}} d(x,a,x')\right)\widehat{\ell}(x,a)$$

As previously stated, the optimization problems can be formulated as different LPs, depending on the choice of the set $\mathcal{I}$. Then the output of the LP, namely d^*, allows to compute the probability function P (played by the algorithm) as:

$$P^{d^*}(x'|x,a) = \frac{d^*(x,a,x')}{\sum\limits_{y\in X_{h(x)+1}} d^*(x,a,y)}$$

In the rest of this section we will use the $\forall h$ term to identify $\forall h \in [0,\dots,H-1]$. We start with the optimization problem for the decision space defined by the module of the difference between transition values for the triple (x,a,x'), namely:

- $\mathcal{I} = \big\{P : |P(x'|x,a) - \overline{P}(x'|x,a)| \leq \epsilon(x,a,x'),\ \ \forall(x,a,x') \in X_h \times A \times X_{h+1}\big\}$

$$\arg\min \sum_{x,a}\left(\sum_{x'\in X_{h(x)+1}} d(x,a,x')\right)\widehat{\ell}(x,a) \tag{5}$$

s.t.

$$\sum_{x\in X_h, a\in A, x'\in X_{h+1}} d(x,a,x') = 1 \qquad \forall h \tag{6}$$

$$\sum_{a\in A, x'\in X_{h+1}} d(x,a,x') = \sum_{x'\in X_{h-1}, a\in A} d(x',a,x) \qquad \forall h, \forall x \in X_h \tag{7}$$

$$d(x,a,x') \leq [\overline{P}(x'|x,a) + \epsilon(x,a,x')] \cdot \sum_{y\in X_{h+1}} d(x,a,y) \qquad \forall h, \forall(x,a,x') \in X_h \times A \times X_{h+1} \tag{8}$$

$$d(x,a,x') \geq [\overline{P}(x'|x,a) - \epsilon(x,a,x')] \cdot \sum_{y\in X_{h+1}} d(x,a,y) \qquad \forall h, \forall(x,a,x') \in X_h \times A \times X_{h+1} \tag{9}$$

$$d(x,a,x') \geq 0 \qquad \forall h, \forall(x,a,x') \in X_h \times A \times X_{h+1} \tag{10}$$

$$d(x,a,x') = 0 \qquad \forall x \in \overline{X}, \forall x' \in X_{h(x)+1}, \forall a \notin \Pi(x) \tag{11}$$

where $\overline{X}$ is the set of visited states, Constraints (6),(7),(10) define a valid occupancy measure, Constraints (8) and (9) define the space of the transition functions and finally Constraint (11) sets to 0 the probability that actions not in Π are played. It easy to check the previous optimization problem is a LP, which can be solved in polynomial time.

We then focus on the case where the distance between transition functions for every tuple (x,a) is computed by the $\|\cdot\|_1$-norm, namely:

- $\mathcal{I} = \big\{P : ||P(\cdot|x,a) - \overline{P}(\cdot|x,a)||_1 \leq \epsilon(x,a),\ \ \forall(x,a) \in X \times A\big\}$

$$\arg\min \sum_{x,a} \left(\sum_{x' \in X_{h(x)+1}} d(x,a,x') \right) \widehat{\ell}(x,a) \tag{12}$$

s.t.

$$\sum_{x \in X_h, a \in A, x' \in X_{h+1}} d(x,a,x') = 1 \qquad \forall h \tag{13}$$

$$\sum_{a \in A, x' \in X_{h+1}} d(x,a,x') = \sum_{x' \in X_{h-1}, a \in A} d(x',a,x) \qquad \forall h, \forall x \in X_h \tag{14}$$

$$d(x,a,x') - \overline{P}(x'|x,a) \cdot \sum_{y \in X_{h+1}} d(x,a,y) \leq \epsilon(x,a,x') \qquad \forall h, \forall (x,a,x') \in X_h \times A \times X_{h+1} \tag{15}$$

$$\overline{P}(x'|x,a) \cdot \sum_{y \in X_{h+1}} d(x,a,y) - d(x,a,x') \leq \epsilon(x,a,x') \qquad \forall h, \forall (x,a,x') \in X_h \times A \times X_{h+1} \tag{16}$$

$$d(x,a,x') \geq 0 \qquad \forall h, \forall (x,a,x') \in X_h \times A \times X_{h+1} \tag{17}$$

$$d(x,a,x') = 0 \qquad \forall x \in \overline{X}, \forall x' \in X_{h(x)+1}, \forall a \notin \Pi(x) \tag{18}$$

$$\sum_{x' \in X_{h+1}} \epsilon(x,a,x') \leq \epsilon(x,a) \cdot \sum_{x' \in X_{h+1}} d(x,a,x') \qquad \forall h, \forall (x,a) \in X_h \times A \tag{19}$$

where Constraints (13),(14),(17) define a valid occupancy measure, Constraints (15), (16) and (19) define the space of the transition functions and finally Constraint (18) sets to 0 the probability that actions not in Π are played. It easy to check the previous optimization problem is a LP, which can be solved in polynomial time.

We conclude the section focusing on the case where transition functions are chosen from a discrete set. We show how the occupancy measure is computed for a fixed transition function P_i. Precisely, the occupancy measure can be obtained with a LP formulation, which implies that $|\mathcal{I}|$ LPs must be solved to obtain the final result. Notice that, since a single LP can be solved in Polynomial time, performing it $|\mathcal{I}|$ times is still polynomial. Moreover, in the discrete case, the value of the occupancy measure given in output by the LP is not necessary; indeed, it is sufficient to obtain the optimal values of the objective function and then to minimize over those values.

- $\mathcal{I}$ is a discrete set

$$\arg\min \sum_{x,a} \left(\sum_{x' \in X_{h(x)+1}} d(x,a,x') \right) \widehat{\ell}(x,a) \tag{20}$$

s.t.

$$\sum_{x \in X_h, a \in A, x' \in X_{h+1}} d(x,a,x') = 1 \qquad \forall h \tag{21}$$

$$\sum_{a \in A, x' \in X_{h+1}} d(x,a,x') = \sum_{x' \in X_{h-1}, a \in A} d(x',a,x) \qquad \forall h, \forall x \in X_h \tag{22}$$

$$d(x,a,x') = P_i(x'|x,a) \sum_{y \in X_{h+1}} d(x,a,y) \qquad \forall h, \forall (x,a,x') \in X_h \times A \times X_{h+1} \tag{23}$$

$$d(x,a,x') \geq 0 \qquad \forall h, \forall (x,a,x') \in X_h \times A \times X_{h+1} \tag{24}$$

$$d(x,a,x') = 0 \qquad \forall x \in \overline{X}, \forall x' \in X_{h(x)+1}, \forall a \notin \Pi(x) \tag{25}$$

where the meaning of the constraints is similar to the ones of the first LP.

In this section, we have shown that the computational complexity of the minimization problem in the *deterministic* setting, namely for Algorithm 2, is polynomial. Notice that, the same result holds in the *stochastic* setting as well. Specifically, it easy to check that by substituting the loss value with its lower-bound, which is not an optimization variable, the same results as in the *deterministic* setting can be obtained.

Omitted Proofs

In the following, we provide the omitted proof of the theorems presented in the main paper, and the related lemmas. For the sake of clarity we name the following subsections as the main paper sections.

Interpreting Our Model as a Lipschitz Bandit

Theorem 2. *Let P, P' be two transition functions, and π an arbitrary Markovian policy. Then,*

$$|J^\pi(P) - J^\pi(P')| \le \frac{H^2}{2} \mathit{TV}(P, P'),$$

where $\mathit{TV}(P,P') := \sum_{x \in X, a \in A} \|P(\cdot|x,a) - P'(\cdot|x,a)\|_1$.

Proof. Let us denote as $d_h^{P,\pi}(\cdot) \in \Delta(X)$ the distribution of states of layer h under configuration P. We have, for every $h > 1$,

$$\begin{aligned}
\|d_h^{P,\pi}(\cdot) - d_h^{P',\pi}(\cdot)\|_1 &= \sum_{x \in X} |d_h^{P,\pi}(x) - d_h^{P',\pi}(x)| \\
&= \sum_{x \in X} \sum_{x_0 \in X, a \in A} |P(x|x_0,a)\pi(a|x_0)d_{h-1}^{P,\pi}(x_0) - P'(x|x_0,a)\pi(a|x_0)d_{h-1}^{P',\pi}(x_0)| \\
&= \sum_{x \in X} \sum_{x_0 \in X, a \in A} \pi(a|x_0)|P(x|x_0,a)d_{h-1}^{P,\pi}(x_0) - P'(x|x_0,a)d_{h-1}^{P',\pi}(x_0)| \\
&\le \sum_{x \in X} \sum_{x_0 \in X, a \in A} \pi(a|x_0)|P(x|x_0,a)d_{h-1}^{P,\pi}(x_0) - P(x|x_0,a)d_{h-1}^{P',\pi}(x_0)| \\
&\qquad\qquad + \pi(a|x_0)|P(x|x_0,a)d_{h-1}^{P',\pi}(x_0) - P'(x|x_0,a)d_{h-1}^{P',\pi}(x_0)| \\
&= \sum_{x \in X} \sum_{x_0 \in X, a \in A} \pi(a|x_0)P(x|x_0,a)|d_{h-1}^{P,\pi}(x_0) - d_{h-1}^{P',\pi}(x_0)| \\
&\qquad\qquad + \pi(a|x_0)d_{h-1}^{P',\pi}(x_0)|P(x|x_0,a) - P'(x|x_0,a)|.
\end{aligned}$$

Here, we can swap the order of the two sums, having, for the first,

$$\begin{aligned}
&\sum_{x_0 \in X, a \in A} \pi(a|x_0)|d_{h-1}^{P,\pi}(x_0) - d_{h-1}^{P',\pi}(x_0)| \sum_{x \in X} P(x|x_0,a) \\
&\qquad = \sum_{x_0 \in X, a \in A} \pi(a|x_0)|d_{h-1}^{P,\pi}(x_0) - d_{h-1}^{P',\pi}(x_0)| \\
&\qquad \le \|d_{h-1}^{P,\pi}(\cdot) - d_{h-1}^{P',\pi}(\cdot)\|_1.
\end{aligned}$$

and, for the second,

$$\begin{aligned}
&\sum_{x_0 \in X, a \in A} \pi(a|x_0)d_{h-1}^{P',\pi}(x_0) \sum_{x \in X} |P(x|x_0,a) - P'(x|x_0,a)| \\
&\qquad = \sum_{x_0 \in X, a \in A} \pi(a|x_0)d_{h-1}^{P',\pi}(x_0)\|P(\cdot|x_0,a) - P'(\cdot|x_0,a)\|_1 \\
&\qquad \le \sum_{x_0 \in X, a \in A} \|P(\cdot|x_0,a) - P'(\cdot|x_0,a)\|_1 = \mathrm{TV}(P,P').
\end{aligned}$$

It follows that $\|d_h^{P,\pi}(\cdot) - d_h^{P,\pi}(\cdot)\|_1 \le \|d_{h-1}^{P,\pi}(\cdot) - d_{h-1}^{P',\pi}(\cdot)\|_1 + \mathrm{TV}(P,P')$. Thus, applying the induction, we get

$$\|d_h^{P,\pi}(\cdot) - d_h^{P',\pi}(\cdot)\|_1 \le h\mathrm{TV}(P,P').$$

Now we focus on the quantity,

$$J^\pi(P) - J^\pi(P') := \sum_{h=1}^{H} \ell(x_h,a_h)\pi(a_h|x_h)d_h^{P,\pi}(x_h).$$

Since the loss is bounded by 1, we get

$$|J^\pi(P) - J^\pi(P')| = \left|\sum_{h=1}^{H} \ell(x_h,a_h)\pi(a_h|x_h)(d_h^{P,\pi}(x_h) - d_h^{P',\pi}(x_h))\right| \le \sum_{h=1}^{H} \|d_h^{P,\pi}(\cdot) - d_h^{P',\pi}(\cdot)\|_1,$$

which, applying the previous relation, is bounded by

$$\frac{H^2}{2}\text{TV}(P, P').$$

□

Deterministic setting

Before being able to prove our main result, let us focus on a simple proposition that will help in the next.

Proposition 8. *Let π_1, π_2 be two policies. Then,*

$$TV(d^{P,\pi_1}, d^{P,\pi_2}) \leq Hd^{P,\pi_2}(\{\pi_1(x) \neq \pi_2(x)\})$$

Proof. Let us suppose $\{\pi_1(x) \neq \pi_2(x)\}$ corresponds to a single state $x_\star$ belonging to layer $h_\star$, in the opposite case we can simply use linearity and sum their visiting distributions. Then,

$$\begin{aligned}\text{TV}(d^{P,\pi_1}, d^{P,\pi_2}) &\leq \sum_{h=1}^{H} \text{TV}(d_h^{P,\pi_1}, d_h^{P,\pi_2}) \\ &= \sum_{h=h_\star}^{H} \text{TV}(d_h^{P,\pi_1}, d_h^{P,\pi_2}).\end{aligned}$$

This is true since, for $h < h_\star$ the effect of $x_\star$ is null. In the opposite case, we have

$$\begin{aligned}\text{TV}(d_h^{P,\pi_1}, d_h^{P,\pi_2}) &= \sup_{S \subset X} \mathbb{P}_{\pi_1}(x_h \in S) - \mathbb{P}_{\pi_2}(x_h \in S) \\ &= \sup_{S \subset X} \mathbb{P}_{\pi_1}(x_h \in S) - \mathbb{P}_{\pi_2}(x_h \in S | x_{h_\star} = x_\star)\mathbb{P}_{\pi_2}(x_{h_\star} = x_\star) \\ &\qquad - \mathbb{P}_{\pi_2}(x_h \in S | x_{h_\star} \neq x_\star)\mathbb{P}_{\pi_2}(x_{h_\star} \neq x_\star),\end{aligned}$$

where the last step holds due to the law of total probabilities. Moreover, under the event $\{s_{h_\star} \neq s_\star\}$, the two process are the same, so that

$$\mathbb{P}_{\pi_1}(x_h \in S | x_{h_\star} \neq s_\star)\mathbb{P}_{\pi_2}(x_{h_\star} \neq x_\star) = \mathbb{P}_{\pi_2}(x_h \in S | x_{h_\star} \neq x_\star)\mathbb{P}_{\pi_2}(x_{h_\star} \neq x_\star).$$

This leads to, for all $h \geq h_\star$,

$$\begin{aligned}\text{TV}(d_h^{P,\pi_1}, d_h^{P,\pi_2}) &= \sup_{S \subset X} \mathbb{P}_{\pi_1}(x_h \in S) - \mathbb{P}_{\pi_2}(x_h \in S) \\ &= \sup_{S \subset X} d^{P,\pi_2}(x_\star)(\mathbb{P}_{\pi_1}(x_h \in S | x_{h_\star} = x_\star) - \mathbb{P}_{\pi_2}(x_h \in S | x_{h_\star} = x_\star)) \\ &\leq d^{P,\pi_2}(x_\star).\end{aligned}$$

Summing over h concludes the proof. □

Theorem 4. *In deterministic settings, Algorithm 2 guarantees a regret upper bound*

$$R_T \leq (H+1)|X|.$$

Proof. **Notation:** Remember that we refer to the loss of a specific state-action pair $x \in X, a \in A$ for a specific episode $t \in [T]$ as $\ell_t(x, a)$. Precisely, in the deterministic setting, $\ell_t(x, a) = \ell(x, a)$.

Since the policy is fixed and deterministic, the loss in a given state is always the same and the dependence on the action can be omitted. For this reason we write

$$\ell(x) := \ell(x, \pi(x)).$$

Using algorithm 2, at any time step we play the configuration $P_t \in \mathcal{I}$ minimizing the following quantity

$$LB_t(P_t) := \min_{P \in \mathcal{I}, \pi \in \Pi} \sum_x d^{P,\pi}(x)\widehat{\ell}(x),$$

where $\widehat{\ell}$ is the loss estimated by the algorithm which, due to the determinism of the loss, is always a lower bound for the true loss. This means being optimistic on the actions of the policy in unknown states, assuming they have loss of 0 (the best possible).

In this way, we have, $LB_t(P) \leq J^\pi(P)$ at any time step t and for any $P \in \mathcal{I}$. From now on, denote as $\overline{X}_t$ the set of unknown state at time t. We can underline some crucial facts about the algorithm:

1. If we have visited all the states we play the optimal configuration $P_t = P_\star$
2. Let us call $\varepsilon_t := J^\pi(P_t) - LB_t(P_t)$. We can note that, at any time step t, we must have

$$\sum_{x \in \overline{X}_t} d^{P_t,\pi}(x) \geq \frac{\varepsilon_t}{H+1}.$$

Indeed,

$$\begin{aligned} J^\pi(P_t) - LB_t(P_t) &= \sum_{x \in X} d^{P_t,\pi}(x)\ell(x) - \sum_{x \in X \setminus \overline{X}_t} d^{P_t,\widehat{\pi}_t}(x)\ell(x) \\ &= \sum_{x \in X} d^{P_t,\pi}(x)\ell(x) - \sum_{x \in X \setminus \overline{X}_t} d^{P_t,\pi}(x)\ell(x) \\ &\quad + \sum_{x \in X \setminus \overline{X}_t} d^{P_t,\pi}(x)\ell(x) - \sum_{x \in X \setminus \overline{X}_t} d^{P_t,\widehat{\pi}}(x)\ell(x) \\ &= \sum_{x \in \overline{X}_t} d^{P_t,\pi}(x)\ell(x) + \sum_{x \in X \setminus \overline{X}_t} (d^{P_t,\pi}(x) - d^{P_t,\widehat{\pi}_t}(x))\ell(x). \end{aligned}$$

Since the loss is in $[0,1]$, the first term is bounded by the sum of the visiting distribution of the unknown states

$$\sum_{\overline{X}_t} d^{P_t,\pi}(x)\ell(x) \leq \sum_{\overline{X}_t} d^{P_t,\pi}(x),$$

while the second one is bounded by $\text{TV}(d^{P_t,\pi_1}, d^{P_t,\pi_2})$, again since the loss is in $[0,1]$. Therefore, we can use proposition 8 to have

$$\text{TV}(d^{P_t,\pi_1}, d^{P_t,\pi_2}) \leq H \sum_{\overline{X}_t} d^{P_t,\pi}(x).$$

Therefore, substituting in the previous formula for the lower bound we get

$$J^\pi(P_t) - LB_t(P_t) \leq \sum_{\overline{X}_t} d^{P_t,\pi}(x) + H \sum_{\overline{X}_t} d^{P_t,\pi}(x) = (H+1) \sum_{\overline{X}_t} d^{P_t,\pi}(x),$$

from which

$$\sum_{\overline{X}_t} d^{P_t,\pi}(x) \geq \frac{J^\pi(P_t) - LB_t(P_t)}{H+1} = \frac{\varepsilon_t}{H+1}.$$

3. Our regret is bounded by $\mathbb{E}\left[\sum_{t=1}^T \varepsilon_t\right]$. Indeed,

$$\begin{aligned} R_T &= \mathbb{E}\left[\sum_{t=1}^T J^\pi(P_t) - J^\pi(P_\star)\right] \\ &\leq \mathbb{E}\left[\sum_{t=1}^T J^\pi(P_t) - LB_t(P_t)\right] \\ &= \mathbb{E}\left[\sum_{t=1}^T \varepsilon_t\right]. \end{aligned}$$

Now, let us define the following sequence of random variables N_t for every $t \in 1, \ldots T$.

$$N_t := \text{number of new states discovered at step } t.$$

With this definition, we can also define the number of states visited up to any time t, which corresponds to the size of $\overline{X}_t$ at that time t,

$$V_t := \sum_{\tau=1}^t N_\tau = |\overline{X}_t|.$$

Also, we will define

$$T_X := \inf\{t \in 1, \dots T : \ V_t = |X|\}.$$

With this definitions, we have indeed

$$\sum_{t=1}^{T_X} N_t = |S| \qquad \text{a.s.} \tag{26}$$

Now, recall that, by points $2, 3$ we have

$$\begin{aligned} R_T &\leq \mathbb{E}\left[\sum_{t=1}^{T} \varepsilon_t\right] \\ &\leq (H+1)\mathbb{E}\left[\sum_{t=1}^{T} \sum_{x \in \overline{X}_t} d^{P_t,\pi}(x)\right] \\ &= (H+1)\mathbb{E}\left[\sum_{t=1}^{T_X} \sum_{x \in \overline{X}_t} d^{P_t,\pi}(x)\right]. \end{aligned}$$

Moreover, since the MDP is assumed without loss of generality to be loop free, the quantity $\sum_{x \in \overline{X}_t} d^{P_t,\pi}(x)$ corresponding to the expected time spent in the set $\overline{X}_t$ at time t, also corresponds to the expected value of the number of states in $\overline{X}_t$ visited, as no state can be visited multiple times in the same episode. This quantity was called N_t in the previous steps. Therefore,

$$\begin{aligned} R_T &\leq (H+1)\mathbb{E}\left[\sum_{t=1}^{T_X} \sum_{x \in \overline{X}_t} d^{P_t,\pi}(x)\right] \\ &= (H+1)\mathbb{E}\left[\sum_{t=1}^{T_X} \mathbb{E}[N_t]\right]. \end{aligned}$$

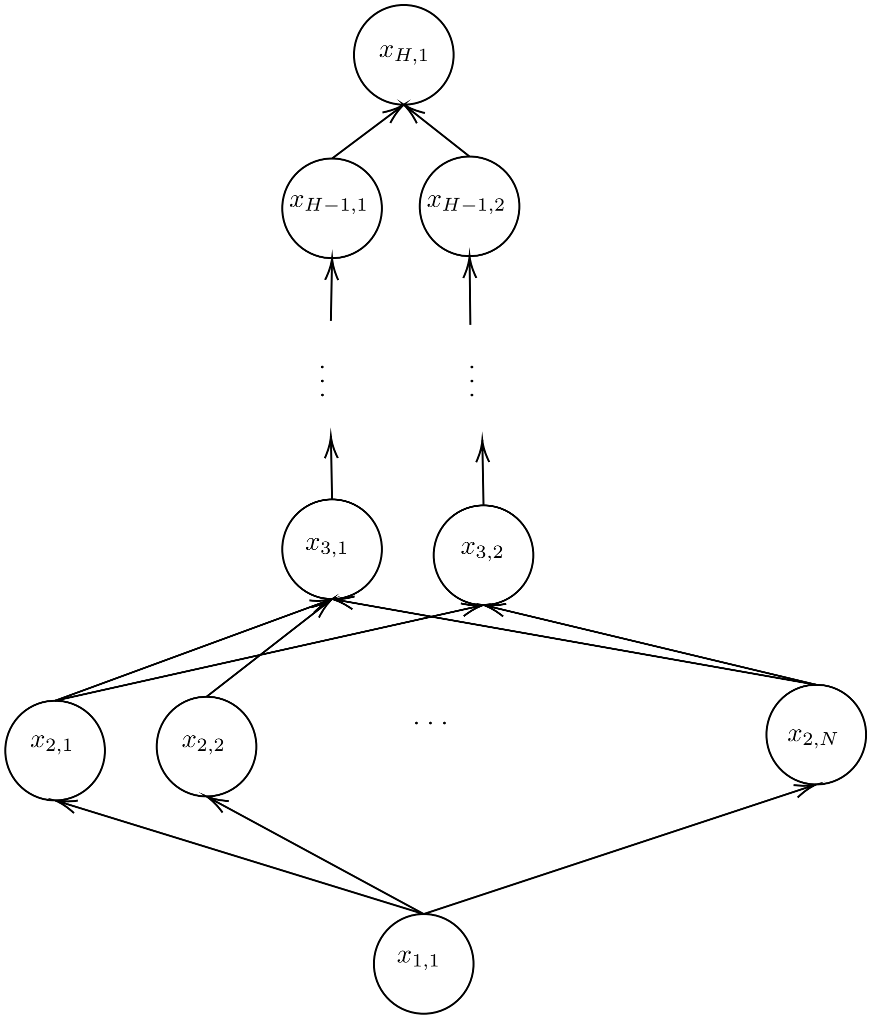
To conclude, we have only to derive a bound on this quantity based on equation (26). Indeed, we have

$$\begin{aligned} R_T &\leq (H+1)\mathbb{E}\left[\sum_{t=1}^{T_X} \mathbb{E}[N_t]\right] \\ &= (H+1)\mathbb{E}\left[\sum_{t=1}^{T_X} N_t\right] \\ &\overset{eq.26}{=} (H+1)|X|. \end{aligned}$$

which concludes the proof.

□

Proof of the lower bound (deterministic setting) . To prove the lower bound, we propose to use a family of MDPs which is represented in the following figure:



Theorem 9. *In deterministic settings, any algorithm achieves a regret of order* $\Omega(H|X|)$.

Proof. We use the family of MDPs defined in the previous figure. Formally, the state space is defined in this way

1. The first and last layers are trivial.
2. The second layer is made by N states.
3. The layers $h = 3, ...H-1$ are made by 2 states.

Instead, the action set corresponds to $\{1, 2\}$. The loss is defined as

$$\ell_h(x,a) = \begin{cases} 1 & x = x_{h,1} \;\; h \in \{3, \dots H-1\} \\ 0 & \text{otherwise} \end{cases}.$$

This means that the loss is only non-zero on the first column (the states of the form $x_{h,1}$ and is constant $+1$). The set $\mathcal{I}$ of possible transition is defined by the set of transitions P satisfying the following conditions

- $P_h(\cdot|x,a)$ is always deterministic
- $h = 2$: for any $x \in X_2, a \in A$

$$P_2(x_{3,1}|x,a) = \begin{cases} 1 & a = 1 \\ 0 & a = 2 \end{cases} \qquad P_2(x_{3,2}|x,a) = \begin{cases} 0 & a = 1 \\ 1 & a = 2 \end{cases}.$$

In other words, at layer 2, the next state is only decided by the action of the agent.

- $h = 3, \dots H-2$ for any $x \in X_2, a \in A$, we have

$$P_h(x_{h+1,1}|x,a) = \begin{cases} 1 & x = x_{h,1} \\ 0 & x = x_{h,2} \end{cases} \qquad P_h(x_{h+1,2}|x,a) = \begin{cases} 0 & x = x_{h,1} \\ 1 & x = x_{h,2} \end{cases}.$$

  Roughly speaking, this tells us that after the second layer, the process proceeds on the same vertical line regardless of the action of the agent.

From these two condition, we can see that the only transition that is not fixed is the one from state $x_{1,1}$ to the second layer, which can be arbitrary, until it is deterministic. This means that we, as configurator can choose arbitrarily the second state of the agent. It is then able to choose the state $x_{3,1}$ or $x_{3,2}$, and proceed on all the states $x_{h,1}$ in the first case or $x_{h,2}$ in the former.

By definition $|\mathcal{I}| = N$, since it corresponds to the possible choice if the state in $h = 2$. At this point, we want to show that for any algorithm there is a problem instance (which in this case is given by the agent policy π, the only element unknown to the configurator) where it cannot achieve expected regret less than $N(H-3)$.

First, note that by Yao's principle it suffices to show that there exist a distribution over the problem instances such that any *deterministic* algorithm suffers at least $N/2(H-3)$ regret when the instance that the algorithm runs on is chosen randomly from the distribution. As distribution of instances we simply choose the uniform distribution over the set Π of the policies π such that

$$\pi_2(2|y) = \begin{cases} 1 & y = x_{2,n} \\ 0 & \text{otherwise} \end{cases} \qquad n \in [1, \dots N].$$

Of course, this set has exactly cardinality N. Indeed, any deterministic algorithm can be viewed as a sequence of permutation of the indexes $1, \dots N$, which are repeated until loss 0 is found. In each round where this is not found, the loss is instead $H-3$. Therefore, by the expected regret of such algorithms can be computed exactly as

$$\mathbb{E}[R_T] \geq (H-3)\frac{N}{2},$$

since, whichever the permutation chosen, the expected order of a random element is $N/2$.

Now, we can rewrite with the substitution $|X| = N + 2(H-2)$, which gives

$$\mathbb{E}[R_T] \geq (H-3)\frac{|X| - 2(H-2)}{2} = \frac{H|X| - 2H^2 - 3|X| + 10H - 12}{2}.$$

□

Stochastic setting

In this section we focus on the more challenging version where the reward is stochastic. Before the main theorem, we have to prove a minor result.

Lemma 2. *Let us consider a sequence of i.i.d. random variables Y_t for $t = 1, \dots T$ of mean μ and bounded in $[0,1]$. For every $\delta > 0$, we have*

$$\mathbb{P}\left(\exists t : \ |\bar{Y}_t - \mu| > \sqrt{\frac{-\log(\delta) + \log(t(t+1))}{2t}}\right) \leq \delta.$$

where

$$\bar{Y}_t := \frac{1}{t}\sum_{i=1}^{t} Y_i.$$

Proof. By Hoeffding's bound, we have, for every t,

$$\begin{aligned}\mathbb{P}\left(\bar{Y}_t - \mu > \sqrt{\frac{-\log(\delta) + \log(t(t+1))}{2t}}\right) &\leq e^{-2t\frac{-\log(\delta)+\log(t(t+1))}{2t}} \\ &= e^{\log(\delta) - \log(t(t+1))} \\ &= \frac{\delta}{t(t+1)}.\end{aligned}$$

Now, we can just use the union bound:

$$\mathbb{P}\left(\exists t:\ \bar{Y}_t - \mu > \sqrt{\frac{-\log(\delta)+\log(t(t+1))}{2t}}\right) \leq \sum_{t=1}^{T}\mathbb{P}\left(\exists t:\ \bar{Y}_t - \mu > \sqrt{\frac{-\log(\delta)+\log(t(t+1))}{2t}}\right)$$
$$\leq \sum_{t=1}^{T}\frac{\delta}{t(t+1)}$$
$$= \delta.$$

At the same way, we can prove

$$\mathbb{P}\left(\exists t:\ \bar{\mu} - Y_t > \sqrt{\frac{-\log(\delta)+\log(t(t+1))}{2t}}\right) \leq \delta,$$

The two results being equivalent to the thesis.

□

Theorem 7. *In the stochastic setting, for the choice $\delta = T^{-1/2}$, Algorithm 3 achieves a regret upper bounded as follows,*

$$R_T = \widetilde{\mathcal{O}}\left(|X|\sqrt{T} + H|X|\right).$$

Proof. **Notation:** Similarly to what has been done before, being $\pi(x)$ deterministic, the dependence on the action can be omitted. For this reason we write

$$\ell(x) := \ell(x, \pi(x)) = \mathbb{E}[\mathcal{L}(x, \pi(x))].$$

At the same way, we will write $\widehat{\ell}_t(x) := \widehat{\ell}_t(x, \pi(x))$ and $\overline{\ell}_t(x) := \overline{\ell}_t(x, \pi(x))$
Our algorithm plays (Line 4), at any time t, the configuration $P \in \mathcal{I}$ minimizing the following lower bound

$$LB_t(P) \ = \operatorname*{arg\,min}_{P\in\mathcal{I},\pi\in\Pi} \sum_{x,a}\left(\sum_{x'\in X_{h(x)+1}} d^{P,\pi}(x,a,x')\right)\widehat{\ell}_t(x,a)$$
$$\widehat{\ell}_t(x,a) = \max\left(0, \overline{\ell}_t(x,a) - \sqrt{\frac{-\log(\delta)+\log(N_t(x)(N_t(x)+1))}{2N_t(x)}}\right).$$

We will call P_t, π_t the couple configuration, policy attaining the minimum.
Define, for every $t = 1, \ldots T$,

$$\varepsilon_t := J^{\pi}(P_t) - LB_t(P_t).$$

(Part 1) *Failure probability.*
Let us note

$$E := \left\{\exists x \in X, t \in [T]:\ |\bar{\ell}_t(x) - \ell(x)| > \sqrt{\frac{-\log(\delta)+\log(N_t(x)(N_t(x)+1))}{2N_t(x)}}\right\},$$

where $N_t(x)$ denotes the number of visits of state x at time t. By lemma 2, we have $\mathbb{P}(E) \leq 2|X|\delta$.

(Part 2) *Decomposition of the regret.* Let us suppose at time t we have pulled a sub-optimal configuration P_t. Assume that we are under the event E^c: we have that all lower bounds are respected, so that at any time step t, $LB_t(P^*) \leq J^{\pi}(P^*)$. This fact allows the following inequality

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T} J^{\pi}(P_t) - J^{\pi}(P_\star)\right]$$
$$\leq \mathbb{E}\left[\sum_{t=1}^{T} J^{\pi}(P_t) - LB_t(P_t)\right]$$
$$= \mathbb{E}\left[\sum_{t=1}^{T} \varepsilon_t\right].$$

In this way we have proved that, under the event E^c, our regret is bounded by $\sum_{t=1}^{T} \varepsilon_t$.
(Part 3) *From regret to visiting state distribution.* By definition, we have at any time t

$$
\begin{aligned}
\varepsilon_t &= J^{\pi}(P_t) - LB_t(P_t) \\
&= \sum_{x \in X} d^{P_t,\pi}(x)\ell(x) - \sum_{x \in X} d^{P_t,\pi}(x)\widehat{\ell}_t(x) \\
&= \sum_{x \in X} d^{P_t,\pi}(x)\ell(x) - \sum_{x \in X} d^{P_t,\pi}(x)\widehat{\ell}_t(x) \\
&\quad + \sum_{x \in X} d^{P_t,\pi}(x)\widehat{\ell}_t(x) - \sum_{x \in X} d^{P_t,\pi}(x)\widehat{\ell}_t(x) \\
&= \sum_{x \in X} d^{P_t,\pi}(x)(\ell(x) - \widehat{\ell}_t(x)) + \sum_{x \in X} (d^{P_t,\pi}(x) - d^{P_t,\pi}(x))\widehat{\ell}_t(x).
\end{aligned}
$$

Note that, under the event E^c, we have, for any $x \in X$,

$$
\begin{aligned}
\ell(x) - \widehat{\ell}_t(x) &= \ell(x) - \bar{\ell}_t(x) + \sqrt{\frac{-\log(\delta) + \log(N_t(x)(N_t(s)+1))}{2t}} \\
&\leq \ell(x) - \bar{\ell}_t(x) + 2\sqrt{\frac{-\log(\delta) + \log(N_t(x)(N_t(x)+1))}{2N_t(x)}} \\
&= \underbrace{2\sqrt{\frac{-\log(\delta) + \log(N_t(x)(N_t(x)+1))}{2N_t(x)}}}_{L(N_t(x),\delta)}.
\end{aligned}
$$

This ensures that

$$
\sum_{x \in X} d^{P_t,\pi}(x)(\ell(x) - \widehat{\ell}_t(x)) \leq \sum_{x \in X} d^{P_t,\pi}(x) L(N_t(x), \delta). \tag{27}
$$

About the second term, we can say that it is bounded by $\text{TV}(d^{P,\pi_1}, d^{P,\pi_2})$, since the reward is in $[0,1]$. Therefore, we can use proposition 8 to have

$$
\sum_{x \in X} (d^{P_t,\pi}(x) - d^{P_t,\pi_t}(x))\widehat{\ell}_t(x) \leq \text{TV}(d^{P,\pi}, d^{P,\pi_t}) \leq H \sum_{\overline{X}_t} d^{P_t,\pi}(x),
$$

where, as in the previous proofs, $\overline{X}_t$ indicates the set of unknown states at time t. If we define the function

$$
G(N_t(x)) = \begin{cases} H & N_t(x) = 0 \\ 0 & N_t(x) \geq 1 \end{cases}
$$

The previous can be rewritten as

$$
\sum_{x \in X} (d^{P_t,\pi}(x) - d^{P_t,\pi_t}(x))\widehat{\ell}_t(x) \leq \sum_{X} d^{P_t,\pi}(x) G(N_t(x)),
$$

which, together with equation (27), gives

$$
\varepsilon_t \leq \sum_{X} d^{P_t,\pi}(x)(L(N_t(x), \delta) + G(N_t(x))).
$$

(Part 4) *Rewriting the regret.* From the previous results, we have

$$
\begin{aligned}
R_T &\leq \mathbb{E}\left[\sum_{t=1}^{T} \varepsilon_t\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \sum_{x \in X} d^{P_t,\pi}(x)(L(N_t(x), \delta) + G(N_t(x)))\right] \\
&= \mathbb{E}\left[\sum_{x \in X} \sum_{t=1}^{T} d^{P_t,\pi}(x)(L(N_t(x), \delta) + G(N_t(x)))\right].
\end{aligned}
$$

Which, noting as $1_{P_t,t}(x)$ the indicator function of state x being visited at step t by configuration P_t, can also be written as

$$R_T \leq \mathbb{E}\left[\sum_X \sum_{t=1}^T d^\pi_{P_t}(x)(L(N_t(x),\delta) + G(N_t(x)))\right]$$
$$= \mathbb{E}\left[\sum_X \sum_{t=1}^T \mathbf{1}_{P_t,t}(x)(L(N_t(x),\delta) + G(N_t(x)))\right]. \tag{28}$$

the last step being valid due to the fact that $\mathbb{E}[\mathbf{1}_{P_t,t}(x)|\mathcal{F}_{t-1}] = d^{P_t,\pi}(x)$, which is true thanks to the loop-free assumption, and the fact that the other two random quantities $N_t(x), P_t$ are $\mathcal{F}_{t-1}-$measurable. Therefore, we need to bound the two sums

$$\sum_{t=1}^T \mathbf{1}_{P_t,t}(x)L(N_t(x),\delta) + \sum_{t=1}^T \mathbf{1}_{P_t,t}(x)G(N_t(x)).$$

(Part 5) *Bounding the two sums.* Due to the fact that $N_t(x) = \sum_{\tau=1}^t \mathbf{1}_{P_t,t}(x)$, we have

$$\sum_{t=1}^T \mathbf{1}_{P_t,t}(x)L(N_t(x),\delta) \leq \sum_{n=1}^T L(n,\delta),$$
$$\sum_{t=1}^T \mathbf{1}_{P_t,t}(x)G(N_t(x)) \leq \sum_{n=1}^T G(n).$$

The second sum is trivial: by definition of G we get exactly H. About the first one we can say that

$$\sum_{n=1}^T L(n,\delta) = \sum_{n=1}^T 2\sqrt{\frac{-\log(\delta) + \log(n(n+1))}{2n}}$$
$$\leq \sum_{n=1}^T 2\sqrt{\frac{-\log(\delta)}{2n}} + 2\sqrt{\frac{\log(n(n+1))}{2n}},$$

by convexity. The first part is

$$\sum_{n=1}^T 2\sqrt{\frac{-\log(\delta)}{2n}} = \sqrt{-2\log(\delta)}\sum_{n=1}^T \frac{1}{\sqrt{n}} \leq \sqrt{-2\log(\delta)}(1 + 2\sqrt{T}).$$

While the second is

$$\sqrt{2}\sum_{n=1}^T \sqrt{\frac{\log(n(n+1))}{n}} \leq \sqrt{2\log(T(T+1))}\sum_{n=1}^T \sqrt{\frac{1}{n}}$$
$$\leq 2\sqrt{\log(T+1)}\sum_{n=1}^T \sqrt{\frac{1}{n}}$$
$$\leq 2\sqrt{\log(T+1)}(1+2\sqrt{T}).$$

Putting all the parts together we have that the sum of all the terms is bounded by

$$H + (2\sqrt{\log(T+1)} + \sqrt{-2\log(\delta)})(1+2\sqrt{T}).$$

(Part 6) *Final considerations.*
As pointed out, the expected regret is bounded by the expected value of the quantity

$$S_T := \sum_{t=1}^T \mathbf{1}_{P_t,t}(x)L(N_t(x),\delta) + \sum_{t=1}^T \mathbf{1}_{P_t,t}(x)G(N_t(x)),$$

that was bounded in the previous step. The expected regret is then bounded as follows, for every $\delta > 0$:

1. Under E, which has probability $2\delta|X|$, the regret is bounded by T.
2. Under E^c, by the previous point

$$\begin{aligned} S_T &\leq \sum_{x \in X} H + (2\sqrt{\log(T+1)} + \sqrt{-2\log(\delta)})(1 + 2\sqrt{T}) \\ &\leq |X|H + |X|(2\sqrt{\log(T+1)} + \sqrt{-2\log(\delta)})(1 + 2\sqrt{T}). \end{aligned}$$

Therefore, choosing $\delta = T^{-1/2}$, we get

$$\begin{aligned} R_T &= T\mathbb{P}(E) + S_T\mathbb{P}(E^c) \\ &\leq 2|X|\sqrt{T} + |X|H + |X|(2\sqrt{\log(T+1)} + \sqrt{\log(T)})(1 + 2\sqrt{T}). \end{aligned}$$

The final expected regret is then bounded by

$$R_T \leq 2|X|\sqrt{T} + |X|H + |X|(2\sqrt{\log(T+1)} + \sqrt{\log(T)})(1 + 2\sqrt{T}).$$

□

Experiments

For the sake of clarity, we report in the followings additional details on the five instances presented in Figures 1,2. Each instance was tested on the MDP presented in Figure 3. We report the original configuration of the MDP (**config 0**) and all the configurations used for the discrete case in Table 1:

State	Action	State	Config 0	Config 1	Config 2	Config 3
S0	A1	S1	0.1	0.9	0.5	0.1
S0	A1	S2	0.9	0.1	0.5	0.9
S0	A0	S1	0.1	0.9	0.5	1.0
S0	A0	S2	0.9	0.1	0.5	0.0
S1	A3	S3	0.1	0.9	0.5	1.0
S1	A3	S4	0.9	0.1	0.5	0.0
S1	A2	S3	1.0	1.0	1.0	1.0
S2	A5	S3	0.1	0.9	0.5	0.1
S2	A5	S4	0.9	0.1	0.5	0.9
S2	A4	S4	1.0	1.0	1.0	1.0
S3	A6	E	1.0	1.0	1.0	1.0
S4	A7	E	1.0	1.0	1.0	1.0

Table 1: Tabular representation of the transition function for each configuration.

- *Instance of Figure 1a*:
  - number of rounds $T = 1000$
  - number of experiments $Exp = 10$
  - arms $n = 4$
  - transition functions described in Table 1
  - loss vector $\boldsymbol{\ell} = [0.58, 0.42, 0.5, 0.4]$,
- *Instance of Figure 1b*:
  - number of rounds $T = 15$
  - number of experiments $Exp = 10$
  - arms $n = 4$
  - transition functions described in Table 1
  - loss vector $\boldsymbol{\ell} = [0.58, 0.42, 0.5, 0.4]$,
- *Instance of Figure 1c*:
  - number of rounds $T = 1000$
  - number of experiments $Exp = 10$

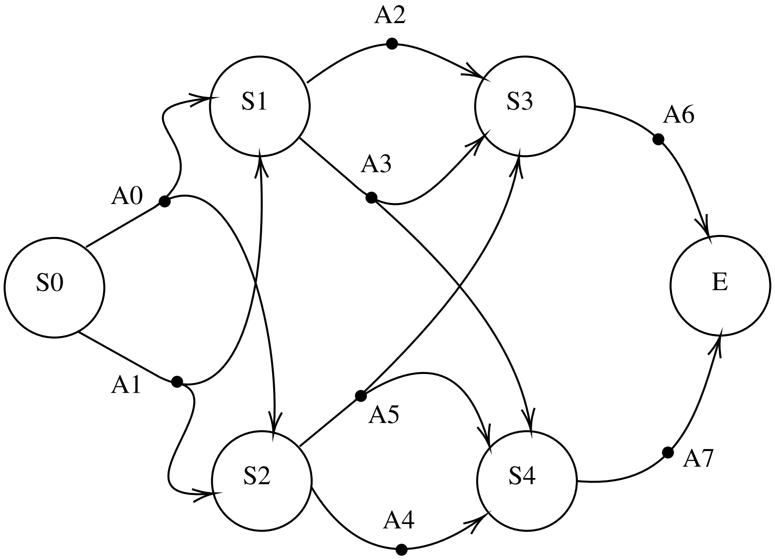


Figure 3: Graphical representation of the MDP used for our experiments

  - arms $n = 4$
  - transition functions $\mathcal{I} = \big\{P : ||P(\cdot|x,a) - \overline{P}(\cdot|x,a)||_1 \leq \epsilon(x,a),\ \ \forall(x,a) \in X \times A\big\}$
  - $\epsilon = 5$
  - loss vector $\boldsymbol{\ell} = [0.58, 0.42, 0.5, 0.4]$,

- *Instance of Figure 2a*:
  - number of rounds $T = 100000$
  - number of experiments $Exp = 10$
  - arms $n = 4$
  - transition functions described in 1
  - mean loss vector $\boldsymbol{\ell} = [0.58, 0.42, 0.5, 0.4]$,
  - unitary variance for each arm
- *Instance of Figure 2b*:
  - number of rounds $T = 100000$
  - number of experiments $Exp = 10$
  - arms $n = 4$
  - transition functions $\mathcal{I} = \big\{P : ||P(\cdot|x,a) - \overline{P}(\cdot|x,a)||_1 \leq \epsilon(x,a),\ \ \forall(x,a) \in X \times A\big\}$
  - $\epsilon = 5$
  - mean loss vector $\boldsymbol{\ell} = [0.58, 0.42, 0.5, 0.4]$,
  - unitary variance for each arm

Training Details In the main paper we have presented five experiments, each corresponding to a different setting. Each experiment is performed with a fixed random seed. The computational time for one experiment depends on the setting. We run the experiments of each setting in parallel with a total computational time of approximately 12 hours.

Compute We run the numerical simulations on a server with the following specifications:

- CPU: `128x Intel(R) Xeon(R) Platinum 8358 CPU @ 2.60GHz`
- RAM: `512,0 GB`
- Operating system: `Ubuntu 20.04.5 LTS`
- System type: `64 bit`

Reproducibility We have performed every experiment with a fixed seed. The seed influences the loss generation by the environment and the transitions to the next states.