Online Learning in Non-Cooperative Configurable Markov Decision Process

Giorgia Ramponi, Alberto Maria Metelli, Alessandro Concetti, Marcello Restelli
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano
Piazza Leonardo da Vinci, 32, 20133, Milano, Italy
{giorgia.ramponi,albertomaria.metelli, marcello.restelli}@polimi.it, alessandro.concetti@mail.polimi.it

Abstract

In the Configurable Markov Decision Processes there are two entities, a Reinforcement Learning agent and a configurator which can modify some parameters of the environment to improve the performance of the agent. What if the configurator does not have the same intentions as the agent? In this paper, we introduce the Non-Cooperative Configurable Markov Decision Process, a framework that allows having two (possibly different) reward functions for the configurator and for the agent. In this setting, we consider an online learning problem, where the configurator has to find the best among a finite set of possible configurations. We propose a learning algorithm to minimize the configurator expected regret, which exploits the structure of the problem. While a naïve application of the UCB algorithm yields a regret that grows indefinitely over time, we show that our approach suffers only bounded regret. Furthermore, we empirically show the performance of our algorithm in simulated domains.

1 Introduction

Reinforcement Learning (RL, Sutton and Barto 1998) has achieved impressive results in several fields of automatic control, including videogames (Mnih et al. 2015), robotics (Peters and Schaal 2008), and autonomous driving (Kiran et al. 2020). The standard RL framework involves an agent whose objective is to maximize the reward collected during its interaction with the environment. However, there are real-world scenarios in which the agent itself or an external supervisor (configurator) can *partially* modify the environment. For example, in a car racing problem, it is possible to modify the car setup to better suit the driver's needs. Recently, the Configurable Markov Decision Processes (Conf-MDPs, Metelli, Mutti, and Restelli 2018) were introduced to model these scenarios and exploit the configuration opportunities. Solving a Conf-MDP consists in simultaneously optimizing a set of environment parameters together with the agent's policy, in order to reach the maximum expected return. This framework has been studied in the discrete and continuous cases (Metelli, Mutti, and Restelli 2018), although the research limited to the case in which the configuration activity aims at maximizing the agent's performance. However, in some cases, the configurator does not know the agent's reward and its intention differs from that of the agent, leading to new appealing scenarios. For instance, imagine we are the owner of a supermarket and we have to decide how to arrange the products on the shelves. Our intention is to increase the final profit of the company; instead, a customer aims at spending the smallest time possible inside the supermarket and buying the indispensable products only. Since we do not know the customer reward function, the only possibility is to try different dispositions and see what is the customers' reaction. But what if we knew what buyers are most interested in? In this case, we can decide *strategically* how to position other products close to the popular ones, to induce the customer in a behavior that is more profitable for us.

In this paper, we introduce the Non-Cooperative Markov Decision Processes (NConf-MDP), a new framework which handles the possibility to have different reward functions for the agent and for the configurator. While Conf-MDP assumes that the configurator acts to help the agent to optimize its expected reward, a NConf-MDP, instead, allows modeling a larger set of scenarios, including all the cases in which agent and configurator display a non-cooperative behavior, modeled by the individual reward functions (Section 3). Obviously, this setting cannot be solved with straightforward application of the algorithms designed for Conf-MDP, that focus on the case in which both entities share the same interests. In fact, if the configurator and the agent optimize separately different objectives they might not converge to an equilibrium strategy. Moreover, accounting of the agent's interest would be, as in Markov games, advantageous for the configurator (Hu and Wellman 2003). In this novel setting, we consider an online learning problem, where the configurator has to select a configuration within a finite set of possible configurations, in order to maximize its own return. This setting can be seen as a *leader-follower* game, in which, the follower (agent) is selfish and optimizes its own reward function, and the leader has to decide the best configuration w.r.t. the best response of the agent. Clearly, in order to adapt its decisions, the configurator has to receive some form of feedback related to the agent's behavior. Specifically, we analyze two settings based on whether the configurator observes just the agent's actions or also a noisy version of the agent's reward function (Section 4). For the two settings, we propose algorithms based on the Optimism in the Face of Uncertainty (OFU, Auer, Cesa-Bianchi, and Fischer 2002)

principle. We show that it is possible to achieve finite expected regret even if the configurator observes the agent's actions only, that scales linearly with the number of admissible configurations (Section 5). Furthermore, we prove that if the configurator observes the noisy agent's reward, under suitable conditions, it is possible to further exploit the *structure* underlying the decision process, removing the dependence on the number of configurations (Section 5). After having revised the literature (Section 6), we provide an experimental evaluation on benchmark domains, inspired to the motivational scenarios of NConf-MDPs, comparing our algorithms with unstructured bandit baselines (Section 7). The proofs of the results presented in the paper are reported in Appendix B.

2 Preliminaries

A *finite-horizon Markov Decision Process* (MDP, Puterman 1994) is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, \mu, r, H)$ where $\mathcal{S}$ is a finite state space ($S = |\mathcal{S}|$), $\mathcal{A}$ is a finite action space ($A = |\mathcal{A}|$), $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition model, which defines the density $p(s'|s, a)$ of state $s' \in \mathcal{S}$ when taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, $\mu : \mathcal{S} \to [0, 1]$ is the initial state distribution, $r : \mathcal{S} \to [0, 1]$ is the reward function, and $H \in \mathbb{N}_{\geq 1}$ is the horizon. A deterministic decision rule $\pi_h : \mathcal{S} \to A$ with $h \in [H]$ prescribes for every states $s \in \mathcal{S}$ an action $\pi_h(s) \in \mathcal{A}$. A deterministic policy $\pi = (\pi_1, \cdots, \pi_H) \in \Pi_D^H$ is a sequence of decision rules, where Π_D^H is the set of deterministic policies.

A *finite-horizon Configurable Markov Decision* Process (Conf-MDP, Metelli, Mutti, and Restelli 2018) is defined as $\mathcal{CM} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mu, r, H)$ and extends the MDP considering a configuration space $\mathcal{P}$ instead a single transition model p. The Q-value of a policy $\pi \in \Pi_D^H$ and configuration $p \in \mathcal{P}$ is the expected sum of the rewards starting from $(s, a) \in \mathcal{S} \times \mathcal{A}$ at step $h \in [H]$:

$$Q_h^{\pi,p}(s, a) = r(s) + \mathop{\mathbb{E}}_{s_{h'} \sim p, \pi} \left[\sum_{h'=h+1}^{H} r(s_{h'}) | s_h = s, a_h = a \right],$$

having denoted with $\mathbb{E}_{s_{h'} \sim p, \pi}$ the expectation w.r.t. the distribution $p(\cdot|s_{h'-1}, \pi_{h'-1}(s_{h'-1}))$. The value function is given by $V_h^{\pi,p}(s) = Q_h^{\pi,p}(s, \pi_h(s))$ and the expected return is defined as $V^{\pi,p} = \mathbb{E}_{s \sim \mu}[V_1^{\pi,p}(s)]$. In a Conf-MDP the goal consists in finding a policy π^* together with an environment configuration p^* so as to maximize the expected return, i.e., $(\pi^*, p^*) \in \arg\max_{\pi \in \Pi_D^H, p \in \mathcal{P}} V^{\pi,p}$.

3 Non-Cooperative Conf-MDPs

The definition of Conf-MDP allows modeling scenarios in which agent and configurator share the same objective, encoded in a single reward function r. In this section, we introduce an extension of this framework to account for the presence of a configurator having interests that might differ from those of the agent.

Definition 3.1. *A Non-Cooperative Configurable Markov Decision Process (NConf-MDP) is defined by a tuple $\mathcal{NCM} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mu, r_c, r_o, H)$, where $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mu, H)$ is a Conf-MDP without reward and $r_c, r_o : \mathcal{S} \to [0, 1]$ are the configurator and agent (opponent) reward functions, respectively.*

Given a policy $\pi = (\pi_h)_{h \in [H]} \in \Pi_D^H$ and a configuration $p \in \mathcal{P}$, for every $(s, a) \in \mathcal{S} \times \mathcal{A}$ and $h \in [H]$ we define the configurator and agent Q-values as:

$$Q_{c,h}^{\pi,p}(s, a) = r_a(s) + \mathop{\mathbb{E}}_{s_{h'} \sim p, \pi} \left[\sum_{h'=h+1}^{H} r_c(s_{h'}) | s_h = s, a_h = a \right],$$

$$Q_{o,h}^{\pi,p}(s, a) = r_o(s) + \mathop{\mathbb{E}}_{s_{h'} \sim p, \pi} \left[\sum_{h'=h+1}^{H} r_o(s_{h'}) | s_h = s, a_h = a \right].$$

We denote with $V_{c,h}^{\pi,p}(s) = Q_{c,h}^{\pi,p}(s, \pi_h(s))$ and $V_{o,h}^{\pi,p} = Q_{o,h}^{\pi,p}(s, \pi_h(s))$ the value functions and with $V_c^{\pi,p} = \mathbb{E}_{s \sim \mu}[V_{c,1}^{\pi,p}(s)]$ and $V_o^{\pi,p} = \mathbb{E}_{s \sim \mu}[V_{o,1}^{\pi,p}(s)]$ the expected returns for the configurator and the agent respectively.

4 Problem Formulation

While for classical Conf-MDPs (Metelli, Mutti, and Restelli 2018) a notion of optimality is straightforward as agent and configurator share the same objective, in a NConf-MDP they can display possibly conflicting interests. We assume a *sequential* interaction between the configurator and the agent, that resembles the leader-follower protocol (Breton, Alj, and Haurie 1988). First, the configurator (leader) selects an environment configuration $p \in \mathcal{P}$ and then the agent (follower) plays a best response policy $\pi_p^* \in \Pi_D^H$, i.e., an optimal policy for the MDP $(\mathcal{S}, \mathcal{A}, p, \mu, r_o, H)$:

$$\pi_p^* \in \mathop{\arg\max}_{\pi \in \Pi_D^H} V_o^{\pi,p}.$$

Before proceeding we make the following assumption.

Assumption 1. *For every environment configuration $p \in \mathcal{P}$, the agent will always play the same best response policy π_p^*. Furthermore, π_p^* is deterministic.*

Requiring that the best response policy is deterministic is justified by the fact that for every MDP there exists at least one deterministic optimal policy (Sutton and Barto 1998; Puterman 1994). The first part of the assumption states that that the agent will react with the same optimal policy π_p^* whenever facing configuration p. This is a common assumption in standard Stackelberg Games (Balcan et al. 2015; Peng et al. 2019; Sessa et al. 2020).

Under Assumption 1, the goal of the configurator is well-defined and consists in finding the configuration $p^* \in \mathcal{P}$ that is optimal under the agent's best response policy:[1]

$$p^* \in \mathop{\arg\max}_{p \in \mathcal{P}} V_c^{\pi_p^*, p}.$$

The configurator knows everything about the NConf-MDP, except for the agent reward function r_o. At each episode $k \in [K]$, the configurator selects a configuration $p_k \in \mathcal{P}$ and observes a trajectory of H steps generated by the agent's best response policy $\pi_{p_k}^*$. We study two types of feedback:

[1] From a game theoretic perspective, the pair $(p^*, \pi_{p^*}^*)$ can be regarded as a Stackelberg equilibrium of the corresponding game (Breton, Alj, and Haurie 1988).

- *Action-feedback* (Af). The configurator observes the states and the actions played by the agent $(s_1, a_1, \ldots, s_{H-1}, a_{H-1}, s_H)$, where $a_h = \pi^*_{p_k,h}(s_h)$.
- *Reward-feedback* (Rf). The configurator observes the states, the actions played by the agent, and a noisy version of the agent reward function $(s_1, \widetilde{r}_1, a_1, \ldots, s_{H-1}, \widetilde{r}_{H-1}, a_{H-1}, s_H, \widetilde{r}_H)$, where $a_h \sim \pi^*_{p_k,h}(s_h)$ and $\widetilde{r}_h$ is sampled from a distribution with mean $r_o(s)$ and support $[0, 1]$.[2]

While the Af is less demanding, the Rf tries to model situations in which the agent's reward is either known under uncertainty or it is obtained in an approximate way through Inverse Reinforcement Learning (Osa et al. 2018).

From an online learning perspective, the goal of the configurator is to minimize the expected regret:

$$\mathbb{E}[\text{Regret}(K)] = \mathbb{E}\left[\sum_{k=1}^{K} \max_{p\in\mathcal{P}} V_c^{\pi^*_p,p} - V_c^{\pi^*_{p_k},p_k}\right]. \quad (1)$$

We assume that the configuration space $\mathcal{P}$ is a finite set made of M stochastic transition models $\mathcal{P} = \{p_1, \ldots, p_M\}$. To lighten the notation, in the following, we will denote with π_i the agent's best response policy to the configuration p_i, i.e., $\pi^*_{p_i}$ and with V^i the configurator expected returned attained with configuration p_i and $\pi^*_{p_i}$, i.e., $V_c^{\pi^*_{p_i},p_i}$. Finally, we denote with $V^* = \max_{i\in[M]} V^i$.

Remark 4.1 (On the optimality of the agent's policy)**.** *In our setting, we assume that the policy the agent plays, at every episode, is an optimal policy. It might be argued that the agent, whenever experiencing a modification of the environment configuration, needs some time to adjust its policy, before reaching optimality. However, in real-world situations, environment configuration and agent learning typically happen on different time scales. Indeed, the configuration changes slowly, giving the agent the time to converge to an optimal policy. For instance, in the supermarket example (Section 1), the time interval between two changes of product disposition might be more than one month, instead a buyer takes less time (few visits) to learn the disposition and their best policy.*

5 Optimistic Configuration Learning

In this section, we present two algorithms for the online learning problem introduced in Section 4. The first algorithm uses only the collected agent decisions to optimistically learn the best configuration (Section 5). In the second algorithm, we use also the noisy reward feedback to construct an algorithm that leverages on the structure that links together all the transition probability models: the agent's reward function r_o. We show that, under suitable assumptions, the regret of the second algorithm removes the dependencies on the number of configurations (Section 5). We conclude the section with a discussion about the considered assumptions and regret guarantees (Section 5).

[2]Clearly, the results we present can be directly extended to subgaussian distributions on the reward.

Action-feedback Optimistic Configuration Learning

We start with the action-feedback (Af) setting in which the configurator observes the agent's actions only. The idea at the basis of the algorithm we propose, *Action-feedback Optimistic Configuration Learning* (AfOCL), is to maintain, for each configuration, a set of *plausible* policies that contains the agent's best response policy. The configurator plays the transition model that maximizes an optimistic approximation of its value function. Specifically, for every $i \in [M]$, $k \in [K]$, and $h \in [H]$ we denote with $\mathcal{A}^i_{k,h}(s) \subseteq \mathcal{A}$ the set of plausible actions in state s at step h for configuration p_i at the beginning of episode k. Since the agent's best response policy π_i is deterministic, if state s is visited at step h before episode k, we know the agent's action in the current model p_i and therefore we set $\mathcal{A}^i_{k,h}(s) = \{\pi_{i,h}(s)\}$, otherwise we have no knowledge and we set $\mathcal{A}^i_{k,h}(s) = \mathcal{A}$. Based on this, we can compute an optimistic approximation $\widetilde{V}^i_{k,h}$ of the configurator value function V^i_h:

$$\widetilde{V}^i_{k,h}(s) = r_c(s) + \max_{a\in\mathcal{A}^i_{k,h}(s)} \sum_{s'\in\mathcal{S}} p_i(s'|s,a)\widetilde{V}^i_{k,h+1}(s'), \quad (2)$$

and $\widetilde{V}^i_{k,H}(s) = r_c(s)$. For visited pairs (s, h) the maximization over the actions reduces to the evaluation of the transition model in the agent's action $\pi_{i,h}(s)$. Clearly, we have that $\widetilde{V}^i_{k,h}(s) \geq V^i_h(s)$ for all $s \in \mathcal{S}$, $h \in [H]$, and $i \in [M]$. Thus, at each episode $k \in [K]$ the configurator plays the transition model p_{I_k} maximizing the optimistic approximation $\widetilde{V}^i_k$:

$$I_k \in \arg\max_{i\in[M]} \widetilde{V}^i_k.$$

The pseudocode of AfOCL is reported in Algorithm 1. The computation of the optimistic approximation $\widetilde{V}^i_{k,h}$ can be simply performed applying a value-iteration-like algorithm (Puterman 1994) that employs the iterate as in Equation (2). Notice that the computational complexity decreases as the number of visited states increases and, in any case, is bounded by that of value iteration $\mathcal{O}\left(HS^2A\right)$. Therefore, the time complexity of AfOCL is $\mathcal{O}\left(KMHS^2A\right)$.

Regret Guarantees In this section, we provide an expected regret bound for the AfOCL algorithm. Since the policy is deterministic, in every episode $k \in [K]$, we acquire the information about which action the agent plays, in the chosen model p_{I_k}, for every visited state. So the main effort is to estimate the agent's policies of every model. In fact, after that, the algorithm will be able to compute the correct expected return for each transition model. However, due to the stochasticity of the models p_i for $i \in [M]$, some states might be visited with low frequency. The following result exploits the determinism of the agent's best response policy to prove that the regret AfOCL suffers is constant, independent on the number of episodes K.

Theorem 5.1 (Regret of AfOCL)**.** *Let* $\mathcal{NCM} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mu, r_c, r_o, H)$ *with* $\mathcal{P} = \{p_1, \ldots, p_M\}$ *be the* M

Algorithm 1 Action-feedback Optimistic Configuration Learning (AfOCL).

1: **Input:** $\mathcal{S}$, $\mathcal{A}$, H, $\mathcal{P} = \{p_1, \dots, p_M\}$
2: Initialize $\mathcal{A}^i_{1,h}(s) = \mathcal{A}$ for all $s \in \mathcal{S}$, $h \in [H]$, and $i \in [M]$
3: **for** episodes $1, 2, \dots, K$ **do**
4: Compute $\widetilde{V}^i_k$ for all $i \in [M]$
5: Play p_{I_k} with $I_k \in \arg\max_{i \in [M]} \widetilde{V}^i_k$
6: Observe $(s_{k,1}, a_{k,1}, \dots, s_{k,H-1}, a_{k,H-1}, s_{k,H})$
7: Compute the plausible actions for all $s \in \mathcal{S}$ and $h \in [H]$:

$$\mathcal{A}^i_{k+1,h}(s) = \begin{cases} \{a_{k,h}\} & \text{if } i = I_k \text{ and } s = s_{k,h} \\ \mathcal{A}^i_{k,h}(s) & \text{otherwise} \end{cases}$$

8: **end for**

finite-horizon MDPs of the problem. The expected regret of AfOCL at every episode $K > 0$ is bounded by:

$$\mathbb{E}[\textit{Regret}(K)] \leq MH^3S^2. \tag{3}$$

The result might be surprising as the regret is constant and independent on the suboptimality gaps between the configurations, i.e., $\Delta_i = V^* - V^i$ for every $i \in [M]$. As supported by intuition, we need to spend more time to discard MDPs that are more similar in performance to the optimal one. Formally, the maximum number of times a suboptimal configuration p_i is played is proportional to $\frac{1}{\Delta_i}$ (and not proportional to $\frac{1}{\Delta_i^2}$ as in standard bandits). This is because the policies are deterministic and, to learn them, we just need *one* visit to the state. More details on the proof are given in the Appendix B.

Reward-feedback Optimistic Configuration Learning

The main drawback of AfOCL is that every transition model is treated separately, preventing from employing the underlying *structure* of the environment. Such a structure is represented by the agent reward function r_o, that is completely ignored in AfOCL. Indeed, if the configurator knew r_o, it could find the optimal configuration with no need of interaction, by simply computing the agent's best response policies. The algorithm we propose in this section, *Reward-feedback Optimistic Configuration Learning* (RfOCL), employs the reward feedback (Rf), i.e., at every interaction the configurator can see also a noisy version of the agent's reward function. The crucial point is that r_o is the same regardless of the chosen configuration and, for this reason, it provides a link between them.

Specifically, for every $k \in [K]$ and $s \in \mathcal{S}$, RfOCL maintains a confidence interval for the agent reward function $\mathcal{R}_k(s) = [\underline{r}_{o,k}(s), \overline{r}_{o,k}(s)]$ obtained using the samples collected up to episode $k-1$ *regardless* of the played configuration. We apply Höeffding inequality to build the confidence intervals obtaining:

$$\widehat{r}_{o,k}(s) \pm \sqrt{\frac{\log(2SHk^2)}{\max\{N_k(s), 1\}}}, \tag{4}$$

where $N_k(s)$ is the number of visits of state s in the first $k-1$ episodes, and $\widehat{r}_{o,k}(s)$ is the sample mean of the observed rewards for state s up to episode k.

Given the estimated reward, for every configuration $i \in [M]$, we can compute a confidence interval for the corresponding agent's Q-values $\mathcal{Q}_{k,h}(s,a) = [\underline{Q}^i_{o,k,h}(s,a), \overline{Q}^i_{o,k,h}(s,a)]$, by simply applying the Bellman equation:

$$\underline{Q}^i_{o,k,h}(s,a) = \underline{r}_{o,k}(s) + \sum_{s' \in \mathcal{S}} p_i(s'|s,a) \max_{a' \in \mathcal{A}} \underline{Q}^i_{o,k,h+1}(s',a'),$$

$$\overline{Q}^i_{o,k,h}(s,a) = \overline{r}_{o,k}(s) + \sum_{s' \in \mathcal{S}} p_i(s'|s,a) \max_{a' \in \mathcal{A}} \overline{Q}^i_{o,k,h+1}(s',a'),$$

and $\underline{Q}^i_{o,k,H}(s,a) = \underline{r}_{o,k}(s)$ and $\overline{Q}^i_{o,k,H}(s,a) = \overline{r}_{o,k}(s)$. If the true reward function belongs to the confidence interval, i.e., $r_o \in \mathcal{R}_k$, then the true Q-value belongs to the corresponding confidence interval, i.e., $Q^i_h \in \mathcal{Q}_{k,h}$. Consequently, we can use $\mathcal{Q}_{k,h}$ to restrict the set of plausible actions in a state *without* actually observing the agent playing the action in that state. Indeed, the plausible actions are those that have an upper Q-value larger than the maximum Q-value lower bound:

$$\widetilde{\mathcal{A}}^i_{k,h}(s) = \left\{ a \in \mathcal{A} : \overline{Q}^i_{o,k,h}(s,a) \geq \max_{a' \in \mathcal{A}} \underline{Q}^i_{o,k,h}(s,a') \right\}. \tag{5}$$

In other words, if the upper Q-value of an action is smaller than the largest lower Q-value, it cannot be the greedy action and it is discarded. Clearly, whenever we observe the agent playing an action in (s,h) we can reduce the plausible actions to the singleton $\{\pi_{i,h}(s)\}$, as in the action-feedback setting (Section 5). Based on this refined definition of plausible actions, we can compute the optimistic estimate $\widetilde{V}^i_{k,h}$ of the configurator value function V^i_h as in Equation (2) and proceed playing the optimistic configuration.

The pseudocode of RfOCL is reported in Algorithm 2. It is worth noting that we need to keep track of the states that have been already visited because for those, we know the agent's action and there is no need to apply Equation (5). This is why we introduce the counts $N_{k,h}(s)$. The computational complexity of an individual iteration of RfOCL is dominated by the value iteration (steps 5 and 9) leading, as for AfOCL, to $\mathcal{O}\left(KMHS^2A\right)$.

Regret Guarantees In this section, we give a regret bound for the RfOCL algorithm. Obviously the same arguments for AfOCL can be also applied for this extended version, and then the regret bound of Theorem B.1 is valid also for RfOCL. Moreover, for this algorithm, we prove that the regret, under certain conditions, does not depend on the number of configurations. In order to prove the result we have to make the following assumption on the NConf-MDP.

Assumption 2. *There exists $\epsilon > 0$ such that:*

$$\min_{i \in [M]} \min_{s \in \mathcal{S}} \max_{h \in [H]} d^i_h(s) \geq \epsilon,$$

where $d^i_h(s)$ is the probability of visiting the state $s \in \mathcal{S}$ at time $h \in [H]$ in configuration p_i under the agent's best response policy π_i.

Algorithm 2 Reward-feedback Optimistic Configuration Learning (RfOCL)

Input: $\mathcal{S}$, $\mathcal{A}$, H, $\mathcal{P} = \{p_1, \dots, p_M\}$
Initialize $\mathcal{A}^i_{1,h}(s) = \mathcal{A}$ for all $s \in \mathcal{S}$, $h \in [H]$, and $i \in [M]$
Initialize $\overline{r}_{o,1}(s) = 1$, $\underline{r}_{o,1}(s) = 0$, and $N_{1,h}(s) = 0$ for all $s \in \mathcal{S}$ and $h \in [H]$
for episodes $1, 2, \dots, K$ **do**
  Compute $\widetilde{V}^i_k$ for all $i \in [M]$
  Play p_{I_k} with $I_k \in \arg\max_{i \in [M]} \widetilde{V}^i_k$
  Observe $(s_{k,1}, \widetilde{r}_{k,1}, a_{k,1}, \dots, s_{k,H-1}, \widetilde{r}_{k,H-1}, a_{k,H-1}, s_{k,H}, \widetilde{r}_{k,H})$
  Compute $\overline{r}_{0,k+1}(s)$, $\underline{r}_{o,k+1}(s)$, and $N_{k+1,h}(s)$ for all $s \in \mathcal{S}$ and $h \in [H]$ using $\widetilde{r}_{k,1} \cdots \widetilde{r}_{k,H}$ as in Equation (4)
  Compute $\underline{Q}^i_{o,k+1,h}(s,a)$, $\overline{Q}^i_{o,k+1,h}(s,a)$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$, $h \in [H]$, and $i \in [M]$
  Compute the plausible actions for all $s \in \mathcal{S}$ and $h \in [H]$:

$$\mathcal{A}^i_{k+1,h}(s) = \begin{cases} \{a_{k,h}\} & \text{if } i = I_k \text{ and } s = s_{k,h} \\ \mathcal{A}^i_{k,h}(s) & \text{if } N_{k,h}(s) > 0 \\ \widetilde{\mathcal{A}}^i_{k+1,h}(s) & \text{otherwise} \end{cases}$$

  with $\widetilde{\mathcal{A}}^i_{k+1,h}(s)$ as in Equation (5).
end for

This assumption involves that in every model $p_i \in \mathcal{P}$ the agent has non-zero probability, in some step h, to visit every state s. This allows shrinking the confidence intervals for the reward of every state in order to estimate correctly the agent's policy, regardless the played configuration. Notice that this assumption is less strict than requiring the ergodicity of the Markov process induced by *any* policy. Under Assumption 2 we can prove the following regret guarantee.

Theorem B.2 (Regret of RfOCL)**.** *Let $\mathcal{NCM} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mu, r_c, r_o, H)$ with $\mathcal{P} = \{p_1, \dots, p_M\}$ be the M finite-horizon MDPs of the problem. Under Assumption 2, the expected regret of RfOCL at every episode $K > 0$ is bounded by:*

$$\mathbb{E}[\textit{Regret(K)}] \leq \min(MH^3S^2, \overline{K}\Delta + \frac{\pi^2}{3}),$$

where $\overline{K}$ is the smallest integer solution of the inequality $\overline{K} \geq 1 + \left(\frac{2H^2S^2 \log(2SH\overline{K}^2)}{2\Delta_Q^2} + \sqrt{\frac{\overline{K}-1}{2} \log(SH\overline{K}^2)} \right) \frac{1}{\epsilon}$, $\Delta = \max_{i \in [M]} \Delta_i$, i.e., the maximum suboptimality gap, and Δ_Q is the minimum positive gap of the agent's Q-values (see Appendix B).

The regret bound removes the dependence on the number of models M, as $\overline{K}$ is clearly independent of M, but it introduces, as expected, a dependence on the minimum visitation probability ϵ. The proof of the result is reported in Appendix B. Since RfOCL exploits additional information compared to AfOCL and the set of plausible actions $\mathcal{A}^i_{k,h}$ of RfOCL are subsets of those of AfOCL, the regret bound AfOCL (Theorem B.1) holds also for RfOCL. Thus, we can take as regret bound for AfOCL the minimum between $\overline{K}\Delta + \frac{\pi^2}{3}$ and MH^3S^2.

Discussion

The two proposed algorithms use different types of feedback acquired by the configuration when interacting with the agent. The second algorithm allows eliminating the dependence on the number of configurations, assuming that the MDP, for each configuration, is ergodic under the optimal policy of the agent. On the other hand, RfOCL is heavier than AfOCL (although the asymptotic complexity is the same) as it requires to compute, for each episode, the optimistic values of the agent Q functions for each model. However, the two algorithms suffer constant regret; this is due to the assumption that the optimal policy of the agent is deterministic. In fact, if we remove this assumption and allow the agent's policy to be stochastic,[3] it is reasonable to believe that the regret AfOCL, suitably modified to maintain confidence intervals for the for the policy, scales logarithmically with K, as in unstructured bandits. We cannot conclude the same for the corresponding adaptation of RfOCL. We conjecture that, under Assumption 2, RfOCL continues to pay constant regret, because it exploits the underlying structure given by the agent's reward function that, allows linking together the different transition models. Thus, when playing any configuration we acquire a finite piece of information that can be shared among all configurations. We leave the investigation of this case as future work.

The online problem that we are facing can be seen as a stochastic multi-armed bandit (Lattimore and Szepesvári 2020), in which the arms are configurations, and the configurator receives a random realization of its expected return at every episode. Thus, it can be solved, in principle, by standard algorithms for bandit problems, such as UCB1 (Auer, Cesa-Bianchi, and Fischer 2002). These algorithms are computationally less demanding, compared to ours, but suffers regret that grows logarithmically, i.e., indefinitely, with the number of episodes. Indeed, they do not exploit either the fact that the agent's policy is deterministic or the structure induced by the agent's reward function.

6 Related Works

The idea of altering the environment dynamics to improve the learning experience of an agent has been exploited before the introduction of Conf-MDPs. *Curriculum learning* (Bengio et al. 2009) provides the agent with a sequence of environments, of increasing difficulty, to shape the learning process with possible benefits on the learning speed e.g., (Ciosek and Whiteson 2017; Florensa et al. 2017). Although the learning process is carried out in a different environment, the configuration is typically performed in simulation only. In the Conf-MDP framework, instead, the configuration opportunities are an *intrinsic* property of the environment (Metelli, Mutti, and Restelli 2018). The initial approaches entitled the agent of the configuration activity and, consequently, this task was totally auxiliary to the agent (Metelli, Mutti, and Restelli 2018; Silva, Melo, and Veloso 2018; **?**). More recently, it has been observed that environment configuration can be actuated even by an external entity, opening new opportunities of

[3] For example, the agent might optimize an entropy-regularized objective (Haarnoja et al. 2018).

application of environment configurability, including settings in which the configurator's interest conflict with those of the agent. For instance, in Metelli, Manneschi, and Restelli (2019) the configurator acts on the environment to induce the agent revealing its capabilities in terms of perception and actuation. Instead, in Gallego, Naveiro, and Insua (2019) a threatener entity can change the transition probabilities either in a stochastic or adversarial manner. More generally, environment configuration carried out by an external entity has been studied in the field of planning as a form of *environment design* (Zhang, Chen, and Parkes 2009). Thus, our NConf-MDP unifies these settings, allowing for arbitrary agent's and configurator's reward functions. An interesting connection is established with the *robust control* literature (Nilim and Ghaoui 2003; Iyengar 2005). Whenever the two reward functions are opposite, i.e., the interaction between the agent and the configuration is fully *competitive*, the resulting equilibrium corresponds to a robust policy. Indeed, while the agent tries to maximize its expected return, the configurator places the agent in the worst possible environment.

The design of our approaches is inspired by classic algorithms based on the OFU principle for stochastic multi-armed bandits (e.g., Lai and Robbins 1985; Auer, Cesa-Bianchi, and Fischer 2002; Garivier and Cappé 2011; Lattimore and Szepesvári 2020) and MDPs (e.g., Auer, Jaksch, and Ortner 2009; Bartlett and Tewari 2012; Azar, Munos, and Kappen 2013). Moreover, our learning setting with reward feedback is related to structured bandits or bandits with correlated arms.[4] Interestingly, for certain structures it is known that bounded regret is achievable (Bubeck, Perchet, and Rigollet 2013; Lattimore and Munos 2014), a property that is enjoyed by both our algorithms. Our setting is also close to the Stochastic Games model, in which two or more agents, acting in an MDP in order to maximize their own reward functions. Recently, the stochastic games framework gains a growing interest (Bai, Jin, and Yu 2020; Bai and Jin 2020; Zhang et al. 2020), especially in the offline setting i.e., we can control all the agents. For this reason these approaches are not applicable to our setting, where we have the control of the configurator only. Although there are also some works which tracts the online setting (Wei, Hong, and Lu 2017; Xie et al. 2020; Tian et al. 2020), where we can control only one agent, all of these algorithms works in the zero-sum setting only.

7 Experiments

In this section, we provide the experimental evaluation of our algorithms on three different domains: Configurable Gridworld (Section 7), Student-Teacher (Section 7), and Configurable Market (Section 7). We compare the algorithms with the standard implementation of UCB1 (Auer, Cesa-Bianchi, and Fischer 2002).The detailed environment description and additional results can be found in Appendix C.

[4] In our case, playing a single configuration provides information about the opponent's reward, which, in turns, provides information about the value of all configurations.

Configurable Gridworld

Configurable Gridworld is a configurable version of a classic 3×3 Gridworld. The starting state of the agent is in the cell $(0, 1)$ and its goal is to minimize the number of steps required to reach the exit, located in the cell $(2, 1)$. Instead, the configurator takes reward 1 when the agent occupies the central cell $(1, 1)$ and 0 otherwise. In a classic gridworld the optimal policy would be trivial, as the agent would proceed straight to the exit. In this Configurable Gridworld, instead, the configurator can set the "power" p of a stochastic obstacle located in the cell $(1, 1)$. In particular, when the agent is in that cell and performs action "go right" to reach the exit, it will hit the obstacle and it will remain in the same position with probability p. The goal of the configurator is to tune this probability in order to keep the agent in the central cell for the maximum number of steps. In practice, this means raising the probability p as much as possible. However, it is easy to prove that if p is too large the agent will learn to avoid the obstacle by passing close to the boundaries, leading to a very poor performance for the configurator. The M configurations differ in the probability p and are obtained by a regular discretization of $[0, 1]$.

The results of the experiments are shown in Figures 1 and 2. In the first experiment, we considered 10 and 30 configurations with a number of episodes $K = 3000$ and horizon $H = 10$. We can see that the two algorithms, AfOCL and RfOCL, suffer constant regret, whereas UCB1 displays a logarithmic regret, as expected. Specifically, RfOCL outperforms AfOCL and stops playing suboptimal configuration in less than 500 episodes in both cases. This can be explained because, being Assumption 2 fulfilled (in fact the agent has the probability 0.1 of failing its action), RfOCL is able to more effectively exploit the underlying structure of the problem. In Figure 2, we present a more extreme case in which we have only three configurations, designed so that the optimal agent's policy generates a non-ergodic Markov chain. In such a case, we violate Assumption 2 and consequently, we observe that AfOCL and RfOCL display a very similar behavior, but still significantly better than UCB1.

Student-Teacher

The Student-Teacher environment models a basic interaction between a student and a teacher. The teacher has a number of exercises available with different difficulty levels and wants to find the optimal sequence of exercises in order to make the student acquire as much knowledge as possible. On the other hand, the student perceives the level of difficulties of the exercises in a different way. The student's goal is to maximize the number of exercises that they know how to solve, and we model this information with an integer between $[0, S]$. Thus, they may decide not to answer and with probability 0.7 they go to an exercise with a lower level of difficulties for the teacher and receives of -1. If it does, they receive a reward that is the level of the "correctness" of the exercise. In Figure 3, we compare our algorithms with UCB1 for different number of configurations $M \in \{40, 60, 100\}$ and horizon $H = 10$. In every run, we construct M random different configurations, that represent the distribution over the next

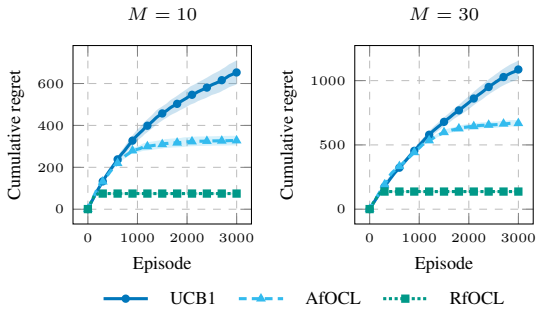


Figure 1: Cumulative regret as a function of the episodes for the Gridworld experiment. 50 runs, 98% c.i.

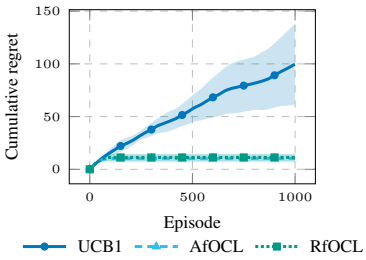


Figure 2: Cumulative regret as a function of the episodes for the Gridworld experiment in the extreme setting. 50 runs, 98% c.i.

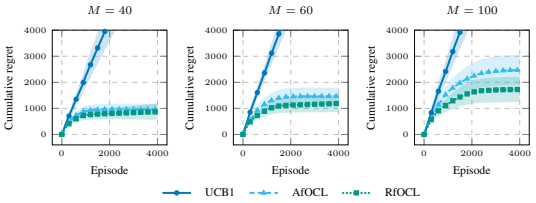


Figure 3: Cumulative regret as a function of the episodes for the Student-Teacher experiment. 50 runs, 98% c.i.

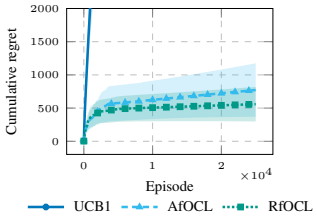


Figure 4: Cumulative regret as a function of the episodes for the Configurable Market experiment. 50 runs, 98% c.i.

exercise, given the current exercise and a positive answer. Moreover, in every run we change the *mildness* of an exercise from the agent's point of view. We observe that both AfOCL and RfOCL suffer significantly less regret compared to UCB1 and tend to converge to a constant, especially with a small number of configurations. It is interesting to observe that, in line with our analysis, the gap between AfOCL and RfOCL appears more evident as the number of configurations grows.

Configurable Market

Configurable Market is a simplified model for a marketplace. The agent, namely the customer, wants to buy a given set of products Q_A in the minimum number of steps. Instead, the configurator has the role to place all the products $Q \supset Q_A$ in the marketplace with the goal to maximize the market's revenue inducing the agent to buy other products in addition to those it would buy. The reward of the configurator is 1 any time the agent passes over a state where a product is placed and 0 in all the other states. Whereas, the reward of the agent is -1 everywhere and gains a bonus of 0.9 when it passes over a state with a product in Q_A. In other words, the products remain fixed in the market and the configurator can change the transition model within a set of random transition models. However, for an abstract point of view this is equivalent to moving the products in the gridworld.[5] In Figure 4, AfOCL and RfOCL are compared against UCB1. The number of configurations is 10, the horizon 15 and the gridworld size is 4×4. In every run, we construct 10 different transition models, which specify the 10 configurations. Also in this experiment, the trend is confirmed since AfOCL and RfOCL outperform UCB1. We observe that the two algorithms, in this environment, behave in a similar way, and this is due to the small number of configurations. However, we can notice RfOCL at the end of the considered episodes approaches the constant regret.

[5]More details are given in Appendix C.

8 Conclusions

In this paper, we have introduced an extension of the Conf-MDP framework to account for possible non-cooperative interaction between the agent and the configurator. We focused on an online learning problem in this new setting, proposing two regret minimization algorithms for identifying the best environment configuration within a finite set based on the principle of optimism in face of uncertainty. We proved that when the agent's policy is deterministic (but the configuration may not) and the configurator observes the agent's actions, it is possible to achieve finite regret that depends linearly on the number of admissible configurations. Furthermore, we illustrated that it is possible to remove this dependence, if the configurator observes a possibly noisy version of the agent's reward and under sufficient regularity conditions on the environment. The experimental evaluation showed that our algorithms display a convergence speed significantly faster compared to UCB1 and RfOCL tends to outperform AfOCL thanks to the exploitation of the additional structure. Future research directions include the extension to the case of stochastic agent's policy and the derivation of specific confidence intervals for the reward function, based on inverse reinforcement learning.

References

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2-3): 235–256.

Auer, P.; Jaksch, T.; and Ortner, R. 2009. Near-optimal regret bounds for reinforcement learning. In *Advances in neural information processing systems*, 89–96.

Azar, M. G.; Munos, R.; and Kappen, H. J. 2013. Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning* 91(3): 325–349.

Bai, Y.; and Jin, C. 2020. Provable Self-Play Algorithms for Competitive Reinforcement Learning. *arXiv preprint arXiv:2002.04017* .

Bai, Y.; Jin, C.; and Yu, T. 2020. Near-Optimal Reinforcement Learning with Self-Play. *arXiv preprint arXiv:2006.12007* .

Balcan, M.-F.; Blum, A.; Haghtalab, N.; and Procaccia, A. D. 2015. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the sixteenth ACM conference on economics and computation*, 61–78.

Bartlett, P. L.; and Tewari, A. 2012. REGAL: A regularization based algorithm for reinforcement learning in weakly communicating MDPs. *arXiv preprint arXiv:1205.2661* .

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In Danyluk, A. P.; Bottou, L.; and Littman, M. L., eds., *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, 41–48. ACM. doi:10.1145/1553374.1553380.

Breton, M.; Alj, A.; and Haurie, A. 1988. Sequential Stackelberg equilibria in two-person games. *Journal of Optimization Theory and Applications* 59(1): 71–97.

Bubeck, S.; Perchet, V.; and Rigollet, P. 2013. Bounded regret in stochastic multi-armed bandits. In *Conference on Learning Theory*, 122–134.

Ciosek, K. A.; and Whiteson, S. 2017. OFFER: Off-Environment Reinforcement Learning. In Singh, S. P.; and Markovitch, S., eds., *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 1819–1825. AAAI Press.

Florensa, C.; Held, D.; Wulfmeier, M.; Zhang, M.; and Abbeel, P. 2017. Reverse Curriculum Generation for Reinforcement Learning. In *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, volume 78 of *Proceedings of Machine Learning Research*, 482–495. PMLR.

Gallego, V.; Naveiro, R.; and Insua, D. R. 2019. Reinforcement Learning under Threats. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 9939–9940. AAAI Press. doi:10.1609/aaai.v33i01.33019939. URL https://doi.org/10.1609/aaai.v33i01.33019939.

Garivier, A.; and Cappé, O. 2011. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*, 359–376.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 1856–1865. PMLR. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

Hu, J.; and Wellman, M. P. 2003. Nash Q-learning for general-sum stochastic games. *Journal of machine learning research* 4(Nov): 1039–1069.

Iyengar, G. N. 2005. Robust Dynamic Programming. *Math. Oper. Res.* 30(2): 257–280. doi:10.1287/moor.1040.0129. URL https://doi.org/10.1287/moor.1040.0129.

Kiran, B. R.; Sobh, I.; Talpaert, V.; Mannion, P.; Sallab, A. A. A.; Yogamani, S. K.; and Pérez, P. 2020. Deep Reinforcement Learning for Autonomous Driving: A Survey. *CoRR* abs/2002.00444.

Lai, T. L.; and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6(1): 4–22.

Lattimore, T.; and Munos, R. 2014. Bounded regret for finite-armed structured bandits. In *Advances in Neural Information Processing Systems*, 550–558.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge University Press.

Metelli, A. M.; Manneschi, G.; and Restelli, M. 2019. Policy Space Identification in Configurable Environments. *CoRR* abs/1909.03984. URL http://arxiv.org/abs/1909.03984.

Metelli, A. M.; Mutti, M.; and Restelli, M. 2018. Configurable Markov Decision Processes. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 3488–3497. PMLR. URL http://proceedings.mlr.press/v80/metelli18a.html.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M. A.; Fidjeland, A.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nat.* 518(7540): 529–533. doi:10.1038/nature14236.

Nilim, A.; and Ghaoui, L. E. 2003. Robustness in Markov Decision Problems with Uncertain Transition Matrices. In Thrun, S.; Saul, L. K.; and Schölkopf, B., eds., *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13,*

2003, Vancouver and Whistler, British Columbia, Canada], 839–846. MIT Press.

Osa, T.; Pajarinen, J.; Neumann, G.; Bagnell, J. A.; Abbeel, P.; and Peters, J. 2018. An Algorithmic Perspective on Imitation Learning. *Found. Trends Robotics* 7(1-2): 1–179. doi:10.1561/2300000053.

Peng, B.; Shen, W.; Tang, P.; and Zuo, S. 2019. Learning optimal strategies to commit to. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2149–2156.

Peters, J.; and Schaal, S. 2008. Reinforcement learning of motor skills with policy gradients. *Neural Networks* 21(4): 682–697. doi:10.1016/j.neunet.2008.02.003.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: John Wiley & Sons, Inc. ISBN 0471619779.

Sessa, P. G.; Bogunovic, I.; Kamgarpour, M.; and Krause, A. 2020. Learning to Play Sequential Games versus Unknown Opponents. *arXiv preprint arXiv:2007.05271* .

Silva, R.; Melo, F. S.; and Veloso, M. 2018. What if the World Were Different? Gradient-Based Exploration for New Optimal Policies. In Lee, D. D.; Steen, A.; and Walsh, T., eds., *GCAI-2018, 4th Global Conference on Artificial Intelligence, Luxembourg, September 18-21, 2018*, volume 55 of *EPiC Series in Computing*, 229–242. EasyChair.

Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. A Bradford book. Bradford Book. ISBN 9780262193986.

Tian, Y.; Wang, Y.; Yu, T.; and Sra, S. 2020. Provably Efficient Online Agnostic Learning in Markov Games. *arXiv preprint arXiv:2010.15020* .

Tirinzoni, A.; Poiani, R.; and Restelli, M. 2020. Sequential Transfer in Reinforcement Learning with a Generative Model. *arXiv preprint arXiv:2007.00722* .

Wei, C.-Y.; Hong, Y.-T.; and Lu, C.-J. 2017. Online reinforcement learning in stochastic games. *arXiv preprint arXiv:1712.00579* .

Xie, Q.; Chen, Y.; Wang, Z.; and Yang, Z. 2020. Learning zero-sum simultaneous-move markov games using function approximation and correlated equilibrium. In *Conference on Learning Theory*, 3674–3682. PMLR.

Zanette, A.; Kochenderfer, M. J.; and Brunskill, E. 2019. Almost Horizon-Free Structure-Aware Best Policy Identification with a Generative Model. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 5625–5634. Curran Associates, Inc.

Zhang, H.; Chen, Y.; and Parkes, D. C. 2009. A General Approach to Environment Design with One Agent. In Boutilier, C., ed., *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, 2002–2014.

Zhang, K.; Kakade, S. M.; Başar, T.; and Yang, L. F. 2020. Model-based multi-agent rl in zero-sum markov games with near-optimal sample complexity. *arXiv preprint arXiv:2007.07461* .

A Notation

$\mathcal{S}$	State space
$\mathcal{A}$	Action space
$\mathcal{P}$	Configuration space
M	Configuration space size
r_o	Agent's reward function
r_c	Configurator's reward function
μ	Initial state distribution
H	Horizon
$Q_{c,h}^{\pi,p}(s,a)$	Configurator's Q-value with policy π and configuration p
$Q_{o,h}^{\pi,p}(s,a)$	Agent's Q-value with policy π and configuration p
$V_{c,h}^{\pi,p}(s)$	Configurator's value function with policy π and configuration p
$V_{o,h}^{\pi,p}(s)$	Agent's value function with policy π and configuration p
$V_c^{\pi,p}$	Configurator's expected return with policy π and configuration p
$V_o^{\pi,p}$	Agent's expected return with policy π and configuration p
$\pi_i = \pi_{p_i}^*$	Agent's best response to configuration p_i
$V^i = V_c^{\pi_{p_i}^*}$	Configurator's expected return with the agent's best response policy $\pi_{p_i}^*$ to configuration p_i
$\widetilde{V}_k^i$	Optimistic configurator's expected return for configuration p_i at episode k
$\widetilde{\pi}_{i,k}$	Estimated agent's best response policy for configuration p_i at episode k
$\Delta_i = V_c^{\pi_{p^*}^*,p^*} - V_c^{\pi_{p_i}^*,p_i}$	Suboptimality gap of the configuration p_i
K	Number of episodes
N_i	Number of time the configuration p_i is played
$N_k(s)$	Number of visit of state s up to episode k
$N_{k,h}^i(s)$	Number of visit of state s at step h up to episode k with configuration p_i
$\underline{r}_{o,k}(s)$	Lower confidence value for the agent's reward
$\overline{r}_{o,k}(s)$	Upper confidence value for the agent's reward
$\widehat{r}_{o,k}(s)$	Sample mean of observed rewards
$\underline{Q}_{o,k,h}^i(s,a)$	Lower confidence value of the agent's Q-function with configuration p_i
$\overline{Q}_{o,k,h}^i(s,a)$	Upper confidence value of the agent's Q-function with configuration p_i
$\mathcal{A}_{k,h}^i(s)$	Set of agent's plausible actions in state s up to episode k
$d_h^i(s)$	Visitation probability the state s at time h with configuration p_i under the agent's best response policy π_i
$\widetilde{d}_h^i(s)$	Visitation probability the state s at time h with configuration p_i under the estimated agent's best response policy $\widetilde{\pi}_{i,k}$

B Missing Proofs

In this appendix, we report the proofs of the results presented in the main paper.

Proofs of Section 5

Lemma B.1. *For every episode $k \in [K]$ and configuration $p_i \in \mathcal{P}$, the difference between the optimistic expected return $\widetilde{V}_k^i$ and the true expected return V^i is bounded by:*

$$\widetilde{V}_k^i - V^i \leq 2H \sum_{s \in \mathcal{S}} \sum_{h=1}^{H-1} d_h^i(s) \mathbb{1}\left\{N_{k,h}^i(s) = 0\right\}. \tag{6}$$

where $N_{k,h}^i(s)$ is the number of times the state $s \in \mathcal{S}$ is visited at step $h \in [H]$ with the configuration $p_i \in \mathcal{P}$ up to episode $k-1$.

Proof. First of all, we denote with $\widetilde{d}_h^i(s)$ the visitation probability of visiting state s at step h under transition model p_i and playing the estimated agent's best response policy $\widetilde{\pi}_i$. Moreover, the visitation probabilities satisfy the following equalities for all $h \geq 2$:

$$\begin{aligned} d_h^i(s) &= \sum_{s' \in \mathcal{S}} p_i(s|s', \pi_{i,h}(s')) d_{h-1}^i(s') \\ \widetilde{d}_h^i(s) &= \sum_{s' \in \mathcal{S}} p_i(s|s', \widetilde{\pi}_{i,h}(s')) \widetilde{d}_{h-1}^i(s'). \end{aligned} \tag{7}$$

Thus, we have:

$$\begin{aligned}
\widetilde{V}_k^i - V^i &= \sum_{s \in \mathcal{S}} \left[\mu(s) r(s) - \mu(s) r(s) + \sum_{h=2}^{H} (\widetilde{d}_h^i(s) - d_h^i(s)) r(s) \right] && (8) \\
&= \sum_{s \in \mathcal{S}} \sum_{h=2}^{H} \left| \widetilde{d}_h^i(s) - d_h^i(s) \right| && (9) \\
&= \sum_{s \in \mathcal{S}} \sum_{h=1}^{H-1} \left| \sum_{s' \in \mathcal{S}} \widetilde{d}_h^i(s') p_i(s|s', \widetilde{\pi}_{i,h}(s')) - d_h^i(s') p_i(s|s', \pi_{i,h}(s')) \right| && (10) \\
&= \sum_{s \in \mathcal{S}} \sum_{h=1}^{H-1} \sum_{s' \in \mathcal{S}} \left| \widetilde{d}_h^i(s') - d_h^i(s') \right| p_i(s|s', \widetilde{\pi}_{i,h}(s')) + d_h^i(s') \left| p_i(s|s', \widetilde{\pi}_{i,h}(s')) - p_i(s|s', \pi_{i,h}(s')) \right| \\
&= \sum_{s' \in \mathcal{S}} \sum_{h=2}^{H-1} \left| \widetilde{d}_h^i(s') - d_h^i(s') \right| + \sum_{s \in \mathcal{S}} \sum_{s' \in \mathcal{S}} \sum_{h=1}^{H-1} d_h^i(s') \left| p_i(s|s', \widetilde{\pi}_{i,h}(s')) - p_i(s|s', \pi_{i,h}(s')) \right| && (11) \\
&= \sum_{H'=2}^{H} \sum_{s \in \mathcal{S}} \sum_{s' \in \mathcal{S}} \sum_{h=1}^{H'-1} d_h^i(s') \left| p_i(s|s', \widetilde{\pi}_{i,h}(s')) - p_i(s|s', \pi_{i,h}(s')) \right| && (12) \\
&\leq H \sum_{s' \in \mathcal{S}} \sum_{h=1}^{H-1} d_h^i(s') \sum_{s \in \mathcal{S}} \left| p_i(s|s', \widetilde{\pi}_{i,h}(s')) - p_i(s|s', \pi_{i,h}(s')) \right| && (13) \\
&\leq 2H \sum_{s' \in \mathcal{S}} \sum_{h=1}^{H-1} \mathbb{1}\left\{N_{k,h}^i(s) = 0\right\} d_h^i(s'), && (14)
\end{aligned}$$

where in line (8) we use the definition of expected return. In line (9) we bound the value of every reward with its maximum value 1. In line (10) we expanded the probability distribution of visiting states using Equations (7). In line (11) we observe that $\widetilde{d}_h^i(s') - d_h^i(s') = \mu(s) - \mu(s) = 0$ to make the first summation start from $h = 2$. In line (12), we apply the recursion with line (9). In line (13), we bound $H' \leq H$ and observe that the outer summation has less than H terms. Finally, in line (14) we upper bound the differences between the two probabilities with 2, and we use the fact that when we have seen a state s with a configuration p_i we have learned its policy (that is deterministic). □

Lemma B.2. *A configuration $p_i \in \mathcal{P}$ is no longer played after episode $k \in [K]$ if for every state $s \in \mathcal{S}$ and $h \in [H]$, with $d_h^i(s) \geq \frac{\Delta_i - c}{2H^2 S}$ we have $N_{k,h}^i(s) > 0$, where $c > 0$ is arbitrary and $\Delta_i = V^* - V^i$.*

Proof. It sufficies to prove that the optimistic expected return $\widetilde{V}_k^i < V^*$, that, in turn, will satisfy $V^* \leq \widetilde{V}_k^{i^*}$:

$$\begin{aligned}
\widetilde{V}_k^i &= V^i + \widetilde{V}_k^i - V^i \\
&\leq V^i + 2H \sum_{s \in \mathcal{S}} \sum_{h=1}^{H-2} d_h^i(s) \mathbb{1}\left\{N_{h,k}^i(s) = 0\right\} && (15) \\
&\leq V^i + 2H^2 S \frac{\Delta_i - c}{2H^2 S} && (16) \\
&= V^i + \Delta_i - c < V^*, && (17)
\end{aligned}$$

where in line (15) we apply Lemma B.1. In line (16) we bound the state visitation probabilities with its maximum value as in the statement hypothesis. In line (17) we use the fact that $\Delta_i = V^* - V_i$. □

Theorem B.1 (Regret of AfOCL)**.** *Let $\mathcal{NCM} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mu, r_c, r_o, H)$ with $\mathcal{P} = \{p_1, \ldots, p_M\}$ be the M finite-horizon MDPs of the problem. The expected regret of AfOCL at every episode $K > 0$ is bounded by:*

$$\mathbb{E}[\textit{Regret}(K)] \leq M H^3 S^2. \tag{18}$$

Proof. We define the regret as:

$$\mathbb{E}[\text{Regret}(K)] = \sum_{i \in [M]: \Delta_i > 0} \Delta_i \, \mathbb{E}[N_i],$$

where N_i is the expected number of times that the algorithm plays model p_i which is not the optimal configuration p_{i^*}. We start bounding for every configuration p_i s.t. $\Delta_i > 0$ the expected value of N_i. We denote with k_l^i the round at which model i is selected for the l-th time:

$$\begin{aligned}
\mathbb{E}[N_i] &\leq \sum_{l=0}^{K} \Pr(N_i \geq l) \\
&\leq \sum_{l=0}^{\infty} \Pr(N_i \geq l) && (19) \\
&\leq \sum_{l=0}^{\infty} \Pr\left(\widetilde{V}_{k_l^i}^i - V^* \geq 0\right) && (20) \\
& && (21)
\end{aligned}$$

In line (19) we extend the sum to ∞. In line (20) we exploit the fact that if configuration i is selected then it must be $\widetilde{V}_{k_l^i}^i \geq \widetilde{V}_{k_l^i}^{i^*}$ and, because of optimism $\widetilde{V}_{k_l^i}^{i^*} \geq V^*$. Then, we observe that for Lemma B.2, if configuration i is played at time k_l^i, then there must exists $s \in \mathcal{S}$ and $h \in [H]$ with $d_h^i(s) \geq \frac{\Delta_i - c}{2H^2 S}$ that is no longer visited. Formally:

$$\begin{aligned}
\mathbb{E}[N_i] &\leq \sum_{l=0}^{\infty} \Pr\left(\widetilde{V}_{k_l^i}^i - V^* \geq 0\right) \\
&\leq \sum_{l=0}^{\infty} \Pr\left(\exists s \in \mathcal{S}, \, \exists h \in [H] \text{ s.t. } d_h^i(s) \geq \frac{\Delta_i - c}{2H^2 S} \; : \; N_{k_l^i, h}^i(s) = 0\right) && (22) \\
&\leq \sum_{l=0}^{\infty} \sum_{s \in \mathcal{S}, h \in [H]: d_h^i(s) \geq \frac{\Delta_i - c}{2H^2 S}} \Pr\left(N_{k_l^i, h}^i(s) = 0\right) && (23) \\
&\leq 1 + SH \sum_{l=1}^{\infty} \left(1 - \frac{\Delta_i - c}{2H^2 S}\right)^{l-1} && (24) \\
&= 1 + SH \frac{1 - \frac{\Delta_i - c}{2H^2 S}}{\frac{\Delta_i - c}{2H^2 S}} \leq \frac{H^3 S^2}{\Delta_i - c}, && (25)
\end{aligned}$$

where, in line (22) we use Lemma B.2. In line (23) we use the union bound over the set employed for existential quantification. In line (24) we bound the probability as $\Pr\left(N_{k_l^i, h}^i(s) = 0\right) = (1 - d_h^i(s))^{l-1}$. In line (25) we use the geometric series properties.

So the expected regret is bounded by:

$$\mathbb{E}[\text{Regret}(K)] = \sum_{i\in[M]:\Delta_i>0} \Delta_i \,\mathbb{E}[N_i] \leq \sum_{i\in[M]:\Delta_i>0} \Delta_i \frac{H^3S^2}{\Delta_i - c} \leq MH^3S^2,$$

having taken the infimum over $c > 0$. □

Proofs of Section 5

In this section, we are going to prove the regret bound RfOCL. In this second algorithm the configurator can observe at every episode also a realization of the agent's reward function. In the following we will show how the algorithm exploits this information under Assumption 2.

We start defining the good events G_k for $k \in [K]$:

$$G_k = \left\{ \exists s \in \mathcal{S} \text{ s.t. } |\widehat{r}_{o,k}(s) - r_o(s)| \leq \sqrt{\frac{\log(2SHk^2)}{2N_k(s)}} \right\}$$

The event G_k means that, at episode $k \in [K]$, the estimated rewards of each state $s \in \mathcal{S}$ are inside the confidence intervals.

Lemma B.3. *For every configuration $p_i \in \mathcal{P}$ and state action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, the difference between the optimistic state-action value function $\overline{Q}^i_{o,k,1}(s,a)$ and the true optimal state-action value function $Q^i_{o,1}(s,a)$ is bounded by:*

$$\overline{Q}^i_{o,k,1}(s,a) - Q^i_{o,1}(s,a) \leq \overline{r}_{o,k}(s) - r_o(s) + \sum_{s'\in\mathcal{S}} \sum_{h=2}^{H} \overline{d}^i_{k,h}(s') \left(\overline{r}_{o,k}(s) - r_o(s)\right),$$

where $\overline{d}^i_{k,h}$ the visitation distribution induced by a greedy policy $\overline{\pi}_{i,k}$ w.r.t. $\overline{Q}^i_{o,k}$. Similarly, the difference between the true optimal state-action value function $Q^i_{o,1}(s,a)$ and the pessimistic state-action value function $\underline{Q}^i_{o,k,1}(s,a)$ is bounded by:

$$Q^i_{o,1}(s,a) - \underline{Q}^i_{o,k,1}(s,a) \leq r_o(s) - \underline{r}_{o,k}(s) + \sum_{s'\in\mathcal{S}} \sum_{h=2}^{H} d^i_{k,h}(s') \left(r_o(s) - \underline{r}_{o,k}(s)\right).$$

Proof. The proof is basically taken from (Zanette, Kochenderfer, and Brunskill 2019; Azar, Munos, and Kappen 2013; Tirinzoni, Poiani, and Restelli 2020):

$$\begin{aligned}
\overline{Q}^i_{o,k,1}(s,a) - Q^i_{o,1}(s,a) &\leq \overline{Q}^i_{o,k,1}(s,a) - Q^{\overline{\pi}_i}_{o,1}(s,a) && (26)\\
&= \overline{r}_{o,k}(s) - r_o(s) + \sum_{s'\in\mathcal{S}} \sum_{h=2}^{H} \overline{d}^i_{k,h}(s') \left(\overline{r}_{o,k}(s) - r_o(s)\right). && (27)
\end{aligned}$$

where line (26) is due to $Q^i_{o,1}(s,a) \geq Q^{\overline{\pi}_{i,k}}_{o,1}(s,a)$, recalling that $Q^i_{o,1}$ is the optimal Q-value for the agent, under configuration p_i and the optimal agent's policy. Line (26) derives form the application of the simulation lemma since $\overline{Q}^i_{o,k,1}(s,a)$ and $Q^{\overline{\pi}_i}_{o,1}(s,a)$ are under the same policy $\overline{\pi}_i$. For the second statement, we proceed analogously by simply observing that $\underline{Q}^i_{o,k,1}(s,a) \leq Q^{\underline{\pi}_{i,k}}_{o,1}(s,a)$ where $\underline{\pi}_{i,k}$ is the greedy policy w.r.t. $\underline{Q}^i_{o,k}(s,a)$. □

Lemma B.4. *If for all $k \in [K]$, the good events G_k hold, for all state-action pairs $(s,a) \in \mathcal{S} \times \mathcal{A}$, $h \in [H]$, and configuration $p_i \in \mathcal{P}$ it holds that:*

$$\overline{Q}^i_{o,k,1}(s,a) - Q^i_{o,1}(s,a) \leq SH\sqrt{\frac{\log(2SHk^2)}{2N_k(s)}},$$

$$Q^i_{o,1}(s,a) - \underline{Q}^i_{o,k,1}(s,a) \leq SH\sqrt{\frac{\log(2SHk^2)}{2N_k(s)}}.$$

Proof. We apply Lemma B.3 and recall that $\overline{r}_{o,k}(s) = \widehat{r}_{o,k}(s) + \sqrt{\frac{\log(2SHk^2)}{2N_k(s)}}$ and $\underline{r}_{o,k}(s) = \widehat{r}_{o,k}(s) - \sqrt{\frac{\log(2SHk^2)}{2N_k(s)}}$. Then, we bound the visitation distribution with 1. □

Lemma B.5. *Let $s \in \mathcal{S}$ be a state with minimum visitation probability $d(s) := \min_{i\in[M]} \max_{h\in[H]} d_h^i(s) > 0$. Then, at episode $k \in [K]$, for every $\delta_k \in (0,1)$, with probability at least $1-\delta_k$ it holds that:*

$$N_k(s) \geq (k-1)d(s) - \sqrt{\frac{k-1}{2}\log\left(\frac{1}{\delta_k}\right)}.$$

Proof. First of all, we define the random variable $N_k^u(s)$ as the count of the visits to state s, where multiple visits in the same episode are considered just once:

$$N_k^u(s) = \sum_{i=1}^{k-1} \mathbb{1}\left\{\exists h \in [H] \,:\, s_{k,h} = s\right\}.$$

Clearly, $N_k^u(s) \leq N_k(s)$ and, consequently, $\mathbb{E}[N_k^u(s)] \leq \mathbb{E}[N_k(s)]$. The expectation of $\mathbb{E}[N_k^u(s)]$ can be bounded as:

$$\begin{aligned}
\mathbb{E}[N_k^u(s)] &= \mathbb{E}\left[\sum_{i=1}^{k-1} \mathbb{1}\left\{\exists h \in [H] \,:\, s_{k,h} = s\right\}\right] \\
&= \sum_{i=1}^{k-1} \Pr\left(\exists h \in [H] \,:\, s_{k,h} = s | p_{I_k}, \pi_{I_k}\right) && (28)\\
&= \sum_{i=1}^{k-1} \Pr\left(\bigcup_{h\in[H]} \{s_{k,h} = s\} | p_{I_k}, \pi_{I_k}\right) && (29)\\
&\geq \sum_{i=1}^{k-1} \max_{h\in[H]} \Pr\left(s_{k,h} = s | p_{I_k}, \pi_{I_k}\right) && (30)\\
&= \sum_{i=1}^{k-1} \max_{h\in[H]} d_h^{I_k}(s) && (31)\\
&\geq (k-1) \min_{i\in[M]} \max_{h\in[H]} d_h^i(s) = (k-1)d(s), && (32)
\end{aligned}$$

where line (28) and line (29) we simply rewrite the expectation as probability. In line (30) we bound the probability of the union with just one term. In line (31) we employ the definition of $d_h^{I_k}(s)$. Finally, in line (32), we take the minimum over I_k. Since $0 \leq N_k^u(s) \leq k-1$, by using Höeffding's inequality we have that with probability at least $1-\delta_k$ it holds that:

$$N_k^u(s) \geq \mathbb{E}[N_k^u(s)] - \sqrt{\frac{k-1}{2}\log\frac{1}{\delta_k}} \geq (k-1)d(s) - \sqrt{\frac{k-1}{2}\log\frac{1}{\delta_k}},$$

having used the lower bound on $\mathbb{E}[N_k^u(s)]$. The result follows from recalling that$\mathbb{E}[N_k^u(s)] \leq \mathbb{E}[N_k(s)]$. □

Lemma B.6. *If for all $k \in [K]$, the good events G_k hold, and for all $s \in \mathcal{S}$ it holds that $\sqrt{\frac{\log(2SHk^2)}{2N_k(s)}} \leq \frac{\Delta_Q - c}{2SH}$, with arbitrary $c > 0$, then for every configuration $p_i \in \mathcal{P}$ we have that $\widetilde{\pi}_i = \pi_i$.*

Proof. Let Δ_Q be the minimum gap between the Q-function in the optimal action and a different action in all transition probabilities $p_i \in \mathcal{P}$:

$$\Delta_Q = \min_{i\in[M]} \min_{s\in\mathcal{S}} \min_{h\in[H]} \left\{ \max_{a\in\mathcal{A}} Q_{o,h}^i(s,a) - \max_{a'\in\mathcal{A}\setminus \arg\max_{a\in\mathcal{A}} Q_{o,h}^i(s,a)} Q_{o,h}^i(s,a') \right\}.$$

For all $s \in \mathcal{S}$ and $h \in [H]$, we denote with $a^* = \arg\max_{a\in\mathcal{A}} Q_{o,h}^i(s,a)$ and we have for all $a \in \mathcal{A} \setminus \{a^*\}$:

$$\begin{aligned}
\overline{Q}_{o,k,h}^i(s,a) - \underline{Q}_{o,k,h}^i(s,a^*) &= \overline{Q}_{o,k,h}^i(s,a) - \underline{Q}_{o,k,h}^i(s,a^*) \pm Q_{o,h}^i(s,a) \pm Q_{o,h}^i(s,a^*) \\
&= \underbrace{\overline{Q}_{o,k,h}^i(s,a) - Q_{o,h}^i(s,a)}_{(A)} + \underbrace{Q_{o,h}^i(s,a^*) - \underline{Q}_{o,k,h}^i(s,a^*)}_{(B)} + \underbrace{Q_{o,h}^i(s,a) - Q_{o,h}^i(s,a^*)}_{(C)} \\
&\leq 2SH\sqrt{\frac{\log(2SHk^2)}{2N_k(s)}} - \Delta_Q \\
&\leq 2SH\frac{\Delta_Q - c}{2SH} - \Delta_Q \leq -c,
\end{aligned}$$

where for (A) and (B) we applied Lemma B.4 and for (C) we used the definition of Δ_Q. We have proved that the lower bound on the Q-value of the optimal action $\underline{Q}^i_{o,k,h}(s,a^*)$ falls above the upper bound on the Q-value of all other actions $\overline{Q}^i_{o,k,h}(s,a)$. Consequently, the greedy action will be properly identified and $\widetilde{\pi}_i = \pi_i$. □

Theorem B.2 (Regret of RfOCL)**.** *Let $\mathcal{NCM} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mu, r_c, r_o, H)$ with $\mathcal{P} = \{p_1, \dots, p_M\}$ be the M finite-horizon MDPs of the problem. Under Assumption 2, the expected regret of RfOCL at every episode $K > 0$ is bounded by:*

$$\mathbb{E}[\textit{Regret(K)}] \leq \min(MH^3S^2, \overline{K}\Delta + \frac{\pi^2}{3}),$$

where $\overline{K}$ is the smallest integer solution of the inequality $\overline{K} \geq 1 + \left(\frac{2H^2S^2\log(2SH\overline{K}^2)}{2\Delta_Q^2} + \sqrt{\frac{\overline{K}-1}{2}\log(SH\overline{K}^2)}\right)\frac{1}{\epsilon}$, $\Delta = \max_{i\in[M]}\Delta_i$, i.e., the maximum suboptimality gap, and Δ_Q is the minimum positive gap of the agent's Q-values (see Appendix B).

Proof. We rewrite the expected regret as follows:

$$\begin{aligned}\mathbb{E}[\text{Regret}(K)] &= \sum_{k=1}^{K}\left(\mathbb{E}[\Delta_{I_k}\mathbb{1}\{G_k\}] + \mathbb{E}[\Delta_{I_k}\mathbb{1}\{\neg G_k\}]\right)\\ &\leq \underbrace{\sum_{k=1}^{K}\mathbb{E}[\Delta_{I_k}|G_k]}_{(A)} + \underbrace{H\sum_{k=1}^{K}\Pr(\neg G_k)}_{(B)},\end{aligned}$$

where we bounded $\Pr(G_k) \leq 1$ in term (A) and Δ_k with its maximum value H in term (B). We start bounding the (B) term:

$$H\sum_{k=1}^{K}\Pr(\neg G_k) = H\sum_{k=1}^{K}\Pr\left(\exists s\in\mathcal{S}\text{ s.t. }|\widehat{r}_{o,k}(s) - r(s)| > \sqrt{\frac{\log(2SHk^2)}{2N_k(s)}}\right) \tag{33}$$

$$\leq H\sum_{k=1}^{K}\sum_{s\in\mathcal{S}}\Pr\left(|\overline{r}_{o,k}(s) - r(s)| > \sqrt{\frac{\log(2SHk^2)}{2N_k(s)}}\right) \tag{34}$$

$$\leq H\sum_{k=1}^{K}\sum_{s\in\mathcal{S}}\frac{1}{SHk^2} \leq \frac{\pi^2}{6}, \tag{35}$$

where line (33) follows from the definition of the good event G_k. Line (34) is a union bound on the states. Line (35) comes from Höeffding's inequality.

For the first term (A) we define the event E_k for all $k \in [K]$:

$$E_k = \left\{\forall s\in\mathcal{S}\,:\,N_k(s) \geq (k-1)d(s) - \sqrt{\frac{k-1}{2}\log\left(SHk^2\right)}\right\}.$$

If this event holds then every state $s \in \mathcal{S}$ is visited at least $(k-1)d(s) - \sqrt{\frac{k}{2}\log\left(SHk^2\right)}$ times, where $d(s)$ is defined as in Lemma **??**.

Considering the term (A), we have:

$$\sum_{k=1}^{K}\mathbb{E}[\Delta_{I_k}|G_k] \leq \underbrace{\sum_{k=1}^{K}\mathbb{E}[\Delta_{I_k}|G_k, E_k]}_{\text{(C)}} + \underbrace{H\sum_{k=1}^{K}\Pr(\neg E_k)}_{\text{(D)}},$$

where we bound the second term with the maximum expected returns-gap with H. We start bounding the second term (D). We apply Lemma B.5 after a union bound over the states:

$$\begin{aligned}H\sum_{k=1}^{K}\Pr(\neg E_k) &= H\sum_{k=1}^{K}\Pr\left(\exists s\in\mathcal{S}: N_k(s) < (k-1)d(s) - \sqrt{\frac{k-1}{2}\log\left(SHk^2\right)}\right)\\ &\leq H\sum_{s\in\mathcal{S}}\sum_{k=1}^{K}\Pr\left(N_k(s) < (k-1)d(s) - \sqrt{\frac{k-1}{2}\log\left(SHk^2\right)}\right)\\ &\leq H\sum_{s\in\mathcal{S}}\sum_{k=1}^{K}\frac{1}{SHk^2} \leq \frac{\pi^2}{6}.\end{aligned}$$

Now it remains to bound the term (C) that, using Lemma B.6, is zero whenever $\sqrt{\frac{\log(2SHk^2)}{2N_k(s)}} \leq \frac{\Delta_Q - c}{2SH}$. Thus, under the events E_k and recalling that under Assumption 2 we have $d(s) \geq \epsilon$, we obtain:

$$\sqrt{\frac{\log(2SHk^2)}{2N_k(s)}} \leq \sqrt{\frac{\log(2SHk^2)}{2(k-1)\epsilon - \sqrt{2(k-1)\log\left(SHk^2\right)}}}.$$

From which, we derive the condition:

$$\overline{K} \geq 1 + \left(\frac{2H^2S^2 \log(2SH\overline{K}^2)}{2(\Delta_Q - c)^2} + \sqrt{\frac{\overline{K}-1}{2}\log(SH\overline{K}^2)} \right) \frac{1}{\epsilon}.$$

Then, we take the infimum over c. Thus, for the term (C), we consider the decomposition:

$$\sum_{k=1}^{K} \mathbb{E}[\Delta_{I_k} | G_k, E_k] \leq \sum_{k=1}^{\overline{K}} \mathbb{E}[\Delta_{I_k} | G_k, E_k] + \sum_{k=\overline{K}+1}^{\infty} \mathbb{E}[\Delta_{I_k} | G_k, E_k] = \overline{K}\Delta + 0,$$

where we bounded $\Delta_{I_k} \leq \Delta$ with $\Delta = \max_{i \in [M]} \Delta_i$. Then the total regret is given by:

$$\mathbb{E}[\text{Regret}(K)] = \overline{K}\Delta + \frac{\pi^2}{3}.$$

□

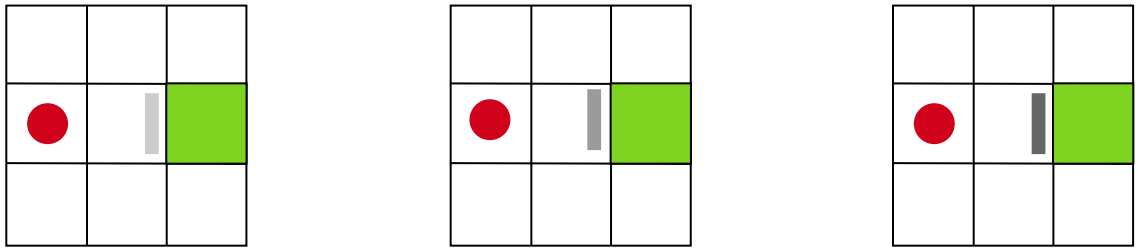
C Experimental Details

In this appendix, we report additional experimental details and results.

Configurable Gridworld

Description In Figure 5 the environment of the Configurable Gridworld is shown. The configurable Gridworld is a 3×3 gridworld with an obstacle in the cell $(2, 2)$, which with a probability p causes the agent action *right* not to be performed. The starting state is in every configuration $(1, 2)$ and the goal state is $(3, 2)$.

Additional Experiments We report additional experiments for the Configurable Gridworld environment. For the Configurable Gridworld with size 3×3, horizon 10, we perform 4 experiments with an increasing number of configurations. Figure 6 shows the results of the experiments. We can notice that with more than 100 configurations AfOCL does not achieve constant regret in 5000 steps, instead RfOCL converges in every experiment.



Configuration #1 Configuration #2 Configuration #3

Figure 5: Configurable Gridworld: from left to right the 3 configurations represent increasing "power" of the obstacle.

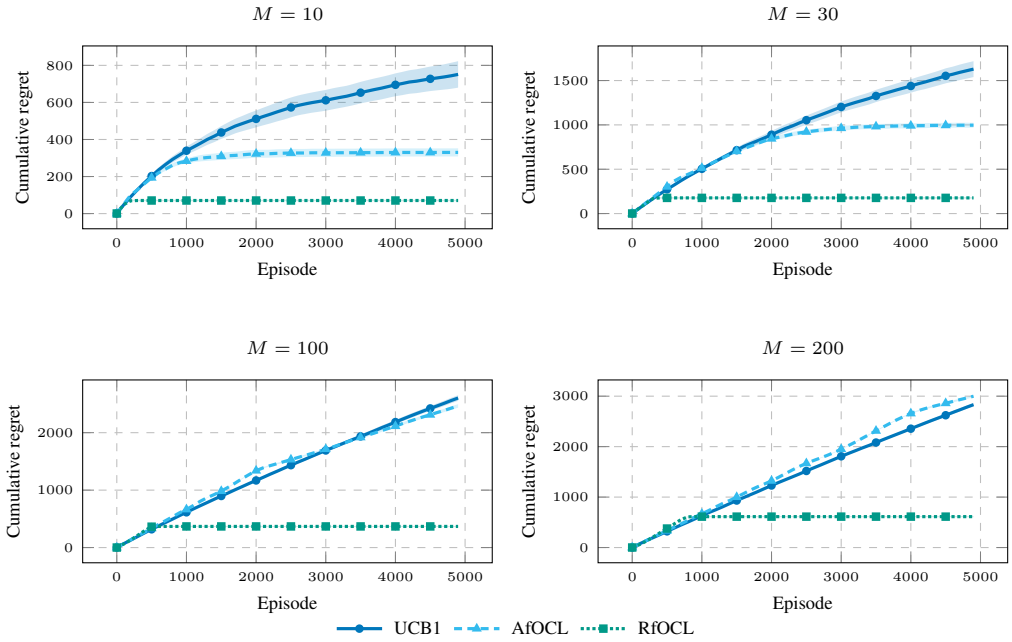
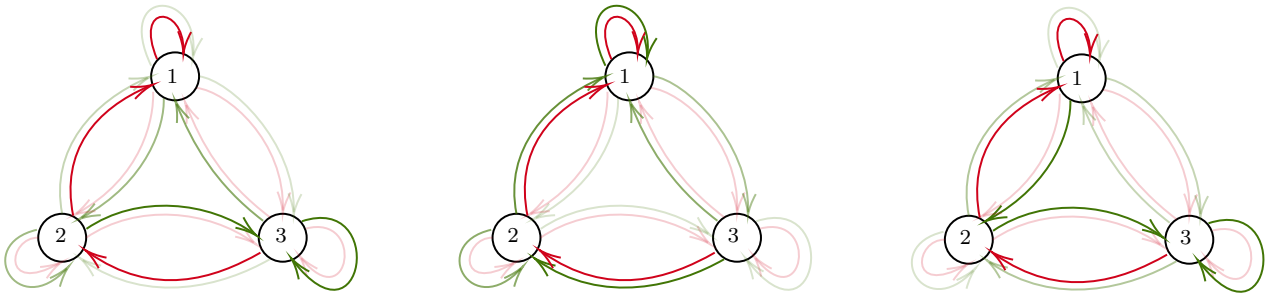


Figure 6: From left up to right down $10, 30, 100, 200$ configurations' number.

Teacher Student

In Figure 7 an illustrative example of the Teacher-Student domain is reported. Right arrows correspond to answer No, and green arrows to answer Yes. The transparency is due to the level of probability of every transition. The configurator can change the transition matrix for the answer Yes, instead the transition matrix for action No is fixed for all the configurations.

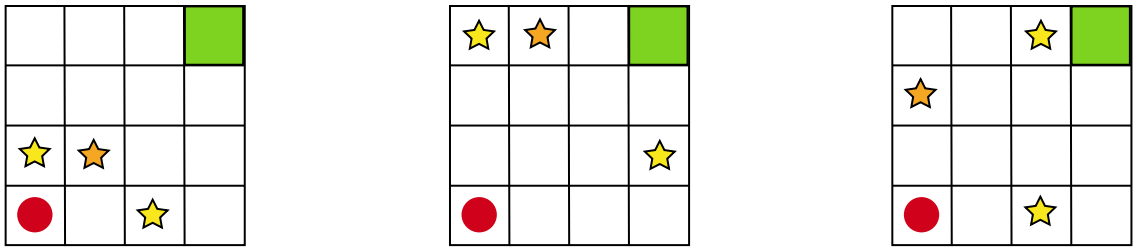


Configuration #1 Configuration #2 Configuration #3

Figure 7: Teacher Student.

Market

In Figure 8 the market domain with 3 different configurations is shown. The market domains consists in $K \times K$ states, where every product is assigned to a specific state. The configurator can change the transition matrix for all the states except for the starting state and the "exit" state. Every different configuration can be thought as shuffling the cells of a gridworld.



Configuration #1 Configuration #2 Configuration #3

Figure 8: Market: the figure shows a 5×5 market. The red state is the starting state, instead the green state is the “end” state. The stars are the product and the orange star is the only product the agent is interested in.