

---

# Selecting the State-Representation in Reinforcement Learning

---

**Odalric-Ambrym Maillard**

INRIA Lille

odalricambrym.maillard@gmail.com

**Rémi Munos**

INRIA Lille

remi.munos@inria.fr

**Daniil Ryabko**

INRIA Lille

daniil@ryabko.net

## Abstract

The problem of selecting the right state-representation in a reinforcement learning problem is considered. Several models (functions mapping past observations to a finite set) of the observations are given, and it is known that for at least one of these models the resulting state dynamics are indeed Markovian. Without knowing neither which of the models is the correct one, nor what are the probabilistic characteristics of the resulting MDP, it is required to obtain as much reward as the optimal policy for the correct model (or for the best of the correct models, if there are several). We propose an algorithm that achieves that, with a regret of order  $T^{2/3}$  where  $T$  is the horizon time.

## 1 Introduction

We consider the problem of selecting the right state-representation in an average-reward reinforcement learning problem. Each state-representation is defined by a model  $\phi_j$  (to which corresponds a state space  $\mathcal{S}_{\phi_j}$ ) and we assume that the number  $J$  of available models is finite and that (at least) one model is a weakly-communicating Markov decision process (MDP). We do not make any assumption at all about the other models. This problem is considered in the general reinforcement learning setting, where an agent interacts with an unknown environment in a single stream of repeated observations, actions and rewards. There are no “resets,” thus all the learning has to be done online. Our goal is to construct an algorithm that performs almost as well as the algorithm that knows both which model is a MDP (knows the “true” model) and the characteristics of this MDP (the transition probabilities and rewards).

Consider some examples that help motivate the problem. The first example is high-level feature selection. Suppose that the space of histories is huge, such as the space of video streams or that of game plays. In addition to these data, we also have some high-level features extracted from it, such as “there is a person present in the video” or “the adversary (in a game) is aggressive.” We know that most of the features are redundant, but we also know that some combination of some of the features describes the problem well and exhibits Markovian dynamics. Given a potentially large number of feature combinations of this kind, we want to find a policy whose average reward is as good as that of the best policy for the right combination of features. Another example is bounding the order of an MDP. The process is known to be  $k$ -order Markov, where  $k$  is unknown but an upper bound  $K \gg k$  is given. The goal is to perform as well as if we knew  $k$ . Yet another example is selecting the right discretization. The environment is an MDP with a continuous state space. We have several candidate quantizations of the state space, one of which gives an MDP. Again, we would like to find a policy that is as good as the optimal policy for the right discretization. This example also opens

the way for extensions of the proposed approach: we would like to be able to treat an infinite set of possible discretization, none of which may be perfectly Markovian. The present work can be considered the first step in this direction.

It is important to note that we do not make any assumptions on the “wrong” models (those that do not have Markovian dynamics). Therefore, we are not able to *test* which model is Markovian in the classical statistical sense, since in order to do that we would need a viable alternative hypothesis (such as, the model is not Markov but is  $K$ -order Markov). In fact, the constructed algorithm never “knows” which model is the right one; it is “only” able to get the same average level of reward as if it knew.

**Previous work.** This work builds on previous work on learning average-reward MDPs. Namely, we use in our algorithm as a subroutine the algorithm UCRL2 of [6] that is designed to provide finite time bounds for undiscounted MDPs. Such a problem has been pioneered in the reinforcement learning literature by [7] and then improved in various ways by [4, 11, 12, 6, 3]; UCRL2 achieves a regret of the order  $DT^{1/2}$  in any weakly-communicating MDP with diameter  $D$ , with respect to the best policy for this MDP. The diameter  $D$  of a MDP is defined in [6] as the expected minimum time required to reach any state starting from any other state. A related result is reported in [3], which improves on constants related to the characteristics of the MDP.

A similar approach has been considered in [10]; the difference is that in that work the probabilistic characteristics of each model are completely known, but the models are not assumed to be Markovian, and belong to a countably infinite (rather than finite) set.

The problem we address can be also viewed as a generalization of the bandit problem (see e.g. [9, 8, 1]): there are finitely many “arms”, corresponding to the policies used in each model, and one of the arms is the best, in the sense that the corresponding model is the “true” one. In the usual bandit setting, the rewards are assumed to be i.i.d. thus one can estimate the mean value of the arms while switching arbitrarily from one arm to the next (the quality of the estimate only depends on the number of pulls of each arm). However, in our setting, estimating the average-reward of a policy requires playing it *many times consecutively*. This can be seen as a bandit problem with dependent arms, with complex costs of switching between arms.

**Contribution.** We show that despite the fact that the true Markov model of states is unknown and that nothing is assumed on the wrong representations, it is still possible to derive a finite-time analysis of the regret for this problem. This is stated in Theorem 1; the bound on the regret that we obtain is of order  $T^{2/3}$ .

The intuition is that if the “true” model  $\phi^*$  is known, but its probabilistic properties are not, then we still know that there exists an optimal control policy that depends on the observed state  $s_{j^*,t}$  only. Therefore, the optimal rate of rewards can be obtained by a clever exploration/exploitation strategy, such as UCRL2 algorithm [6]. Since we do not know in advance which model is a MDP, we need to explore them all, for a sufficiently long time in order to estimate the rate of rewards that one can get using a good policy in that model.

**Outline.** In Section 2 we introduce the precise notion of model and set up the notations. Then we present the proposed algorithm in Section 3; it uses UCRL2 of [6] as a subroutine and selects the models  $\phi$  according to a penalized empirical criterion. In Section 4 we discuss some directions for further development. Finally, Section 5 is devoted to the proof of Theorem 1.

## 2 Notation and definitions

We consider a space of observations  $\mathcal{O}$ , a space of actions  $\mathcal{A}$ , and a space of rewards  $\mathcal{R}$  (all assumed to be Polish). Moreover, we assume that  $\mathcal{A}$  is of finite cardinality  $A \stackrel{\text{def}}{=} |\mathcal{A}|$  and that  $0 \in \mathcal{R} \subset [0, 1]$ . The set of histories up to time  $t$  for all  $t \in \mathbb{N} \cup \{0\}$  will be denoted by  $\mathcal{H}_{<t} \stackrel{\text{def}}{=} \mathcal{O} \times (\mathcal{A} \times \mathcal{R} \times \mathcal{O})^{t-1}$ , and we define the set of all possible histories by  $\mathcal{H} \stackrel{\text{def}}{=} \bigcup_{t=0}^{\infty} \mathcal{H}_{<t}$ .

**Environments.** For a Polish  $\mathcal{X}$ , we Denote by  $\mathcal{P}(\mathcal{X})$  the set of probability distributions over  $\mathcal{X}$ . Define an environment to be a mapping from the set of histories  $\mathcal{H}$  to the set of functions that map any action  $a \in \mathcal{A}$  to a probability distribution  $\nu_a \in \mathcal{P}(\mathcal{R} \times \mathcal{O})$  over the product space of rewards and observations.

We consider the problem of reinforcement learning when the learner interacts with some *unknown* environment  $e^*$ . The interaction is sequential and goes as follows: first some  $h_{<1} = \{o_0\}$  is generated according to  $\iota$ , then at time step  $t > 0$ , the learner chooses an action  $a_t \in \mathcal{A}$  according to the current history  $h_{<t} \in \mathcal{H}_{<t}$ . Then a couple of reward and observations  $(r_t, o_t)$  is drawn according to the distribution  $(e^*(h_{<t}))_{a_t} \in \mathcal{P}(\mathcal{R} \times \mathcal{O})$ . Finally,  $h_{<t+1}$  is defined by the concatenation of  $h_{<t}$  with  $(a_t, r_t, o_t)$ . With these notations, at each time step  $t > 0$ ,  $o_{t-1}$  is the last observation given to the learner before choosing an action,  $a_t$  is the action output at this step, and  $r_t$  is the immediate reward received after playing  $a_t$ .

**State representation functions (models).** Let  $\mathcal{S} \subset \mathbb{N}$  be some finite set; intuitively, this has to be considered as a set of states. A *state representation* function  $\phi$  is a function from the set of histories  $\mathcal{H}$  to  $\mathcal{S}$ . For a state representation function  $\phi$ , we will use the notation  $\mathcal{S}_\phi$  for its set of states, and  $s_{t,\phi} := \phi(h_{<t})$ .

In the sequel, when we talk about a Markov decision process, it will be assumed to be *weakly communicating*, which means that for each pair of states  $u_1, u_2$  there exists  $k \in \mathbb{N}$  and a sequence of actions  $\alpha_1, \dots, \alpha_k \in \mathcal{A}$  such that  $P(s_{k+1,\phi} = u_2 | s_{1,\phi} = u_1, a_1 = \alpha_1 \dots a_k = \alpha_k) > 0$ . Having that in mind, we introduce the following definition.

**Definition 1** *We say that an environment  $e$  with a state representation function  $\phi$  is Markov, or, for short, that  $\phi$  is a Markov model (of  $e$ ), if the process  $(s_{t,\phi}, a_t, r_t), t \in \mathbb{N}$  is a (weakly communicating) Markov decision process.*

For example, consider a state-representation function  $\phi$  that depends only on the last observation, and that partitions the observation space into finitely many cells. Then an environment is Markov with this representation function if the probability distribution on the next cells only depends on the last observed cell and action. Note that there may be many state-representation functions with which an environment  $e$  is Markov.

### 3 Main results

Given a set  $\Phi = \{\phi_j; j \leq J\}$  of  $J$  state-representation functions (models), one of which being a Markov model of the unknown environment  $e^*$ , we want to construct a strategy that performs nearly as well as the best algorithm that knows which  $\phi_j$  is Markov, and knows all the probabilistic characteristics (transition probabilities and rewards) of the MDP corresponding to this model. For that purpose we define the regret of any strategy at time  $T$ , like in [6, 3], as

$$\Delta(T) \stackrel{\text{def}}{=} T\rho^* - \sum_{t=1}^T r_t,$$

where  $r_t$  are the rewards received when following the proposed strategy and  $\rho^*$  is the average optimal value in the best Markov model, i.e.,  $\rho^* = \lim_T \frac{1}{T} \mathbb{E}(\sum_{t=1}^T r_t(\pi^*))$  where  $r_t(\pi^*)$  are the rewards received when following the optimal policy for the best Markov model. Note that this definition makes sense since when the MDP is weakly communicating, the average optimal value of reward does not depend on the initial state. Also, one could replace  $T\rho^*$  with the expected sum of rewards obtained in  $T$  steps (following the optimal policy) at the price of an additional  $O(\sqrt{T})$  term.

In the next subsection, we describe an algorithm that achieves a sub-linear regret of order  $T^{2/3}$ .

#### 3.1 Best Lower Bound (BLB) algorithm

In this section, we introduce the Best-Lower-Bound (BLB) algorithm, described in Figure 1.

The algorithm works in stages of doubling length. Each stage consists in 2 phases: an exploration and an exploitation phase. In the exploration phase, BLB plays the UCRL2 algorithm on each model  $(\phi_j)_{1 \leq j \leq J}$  successively, as if each model  $\phi_j$  was a Markov model, for a fixed number  $\tau_{i,1}, J$  of rounds. The exploitation part consists in selecting first the model with highest lower bound, according to the empirical rewards obtained in the previous exploration phase. This model is initially selected for the same time as in the exploration phase, and then a test decides to either continue playing this model (if its performance during exploitation is still above the corresponding lower bound, i.e. if the rewards obtained are still at least as good as if it was playing the best model). If it does not pass the test, then another model (with second best lower-bound) is select and played, and so on. Until the exploitation phase (of fixed length  $\tau_{i,2}$ ) finishes and the next stage starts.

Parameters:  $f, \delta$   
For each stage  $i \geq 1$  do  
Set the total length of stage  $i$  to be  $\tau_i := 2^i$ .  
1. Exploration. Set  $\tau_{i,1} = \tau_i^{2/3}$ . For each  $j \in \{1, \dots, J\}$  do  
– Run UCRL2 with parameter  $\delta_i(\delta)$  defined in (1) using  $\phi_j$  during  $\tau_{i,1,J}$  steps: the state space is assumed to be  $\mathcal{S}_{\phi_j}$  with transition structure induced by  $\phi_j$ .  
– Compute the corresponding average empirical reward  $\widehat{\mu}_{i,1}(\phi_j)$  received during this exploration phase.  
2. Exploitation. Set  $\tau_{i,2} = \tau_i - \tau_{i,1}$  and initialize  $\mathcal{J} := \{1, \dots, J\}$ .  
While the current length of the exploitation part is less than  $\tau_{i,2}$  do  
– Select  $\widehat{j} = \underset{j \in \mathcal{J}}{\operatorname{argmax}} \widehat{\mu}_{i,1}(\phi_j) - 2B(i, \phi_j, \delta)$  (using (3)).  
– Run UCRL2 with parameter  $\delta_i(\delta)$  using  $\phi_{\widehat{j}}$ : update at each time step  $t$  the current average empirical reward  $\widehat{\mu}_{i,2,t}(\phi_{\widehat{j}})$  from the beginning of the run. Provided that the length of the current run is larger than  $\tau_{i,1,J}$ , do the test  

$$\widehat{\mu}_{i,2,t}(\phi_{\widehat{j}}) \geq \widehat{\mu}_{i,1}(\phi_{\widehat{j}}) - 2B(i, \phi_{\widehat{j}}, \delta).$$
  
– If the test fails, then stop UCRL2 and set  $\mathcal{J} := \mathcal{J} \setminus \{\widehat{j}\}$ . If  $\mathcal{J} = \emptyset$  then set  $J := \{1, \dots, J\}$ .

Figure 1: The Best-Lower-Bound selection strategy.

The length of stage  $i$  is fixed and defined to be  $\tau_i \stackrel{\text{def}}{=} 2^i$ . Thus for a total time horizon  $T$ , the number of stages  $I(T)$  before time  $T$  is  $I(T) \stackrel{\text{def}}{=} \lfloor \log_2(T + 1) \rfloor$ . Each stage  $i$  (of length  $\tau_i$ ) is further decomposed into an exploration (length  $\tau_{i,1}$ ) and an exploitation (length  $\tau_{i,2}$ ) phases.

**Exploration phase.** All the models  $\{\phi_j\}_{j \leq J}$  are played one after another for the same amount of time  $\tau_{i,1,J} \stackrel{\text{def}}{=} \frac{\tau_{i,1}}{J}$ . Each episode  $1 \leq j \leq J$  consists in running the UCRL2 algorithm using the model of states and transitions induced by the state-representation function  $\phi_j$ . Note that UCRL2 does not require the horizon  $T$  in advance, but requires a parameter  $p$  in order to ensure a near optimal regret bound with probability higher than  $1 - p$ . We define this parameter  $p$  to be  $\delta_i(\delta)$  in stage  $i$ , where

$$\delta_i(\delta) \stackrel{\text{def}}{=} (2^i - (J^{-1} + 1)2^{2i/3} + 4)^{-1} 2^{-i+1} \delta. \quad (1)$$

The average empirical reward received during each episode is written  $\widehat{\mu}_{i,1}(\phi_j)$ .

**Exploitation phase.** We use the empirical rewards  $\widehat{\mu}_{i,1}(\phi_j)$  received in the previous exploration part of stage  $i$  together with a confidence bound in order to select the model to play. Moreover, a model  $\phi$  is no longer run for a fixed period of time (as in the exploration part of stage  $i$ ), but for a period  $\tau_{i,2}(\phi)$  that depends on some test; we first initialize  $\mathcal{J} := \{1, \dots, J\}$  and then choose

$$\widehat{j} \stackrel{\text{def}}{=} \underset{j \in \mathcal{J}}{\operatorname{argmax}} \widehat{\mu}_{i,1}(\phi_j) - 2B(i, \phi_j, \delta), \quad (2)$$

where we define

$$B(i, \phi, \delta) \stackrel{\text{def}}{=} 34f(\tau_i - 1 + \tau_{i,1}) |\mathcal{S}_\phi| \sqrt{\frac{A \log\left(\frac{\tau_{i,1,J}}{\delta_i(\delta)}\right)}{\tau_{i,1,J}}}, \quad (3)$$

where  $\delta$  and the function  $f$  are parameters of the BLB algorithm. Then UCRL2 is played using the selected model  $\phi_{\widehat{j}}$  for the parameter  $\delta_i(\delta)$ . In parallel we test whether the average empirical reward we receive during this exploitation phase is high enough; at time  $t$ , if the length of the current episode is larger than  $\tau_{1,i,J}$ , we test if

$$\widehat{\mu}_{i,2,t}(\phi_{\widehat{j}}) \geq \widehat{\mu}_{i,1}(\phi_{\widehat{j}}) - 2B(i, \phi_{\widehat{j}}, \delta). \quad (4)$$

If the test is positive, we keep playing UCRL2 using the same model. Now, if the test fails, then the model  $\widehat{j}$  is discarded (until the end of stage  $i$ ) i.e. we update  $\mathcal{J} := \mathcal{J} \setminus \{\widehat{j}\}$  and we select a new one according to (2). We repeat those steps until the total time  $\tau_{i,2}$  of the exploitation phase of stage  $i$  is over.

**Remark** Note that the model selected for exploitation in (2) is the one that has the best lower bound. This is a pessimistic (or robust) selection strategy. We know that if the right model is selected, then with high probability, this model will be kept during the whole exploitation phase. If this is not the right model, then either the policy provides good rewards and we should keep playing it, or it does not, in which case it will not pass the test (4) and will be removed from the set of models that will be exploited in this phase.

### 3.2 Regret analysis

**Theorem 1 (Main result)** *Assume that a finite set of  $J$  state-representation functions  $\Phi$  is given, and there exists at least one function  $\phi^* \in \Phi$  such that with  $\phi^*$  as a state-representation function the environment is a Markov decision process. If there are several such models, let  $\phi^*$  be the one with the highest average reward of the optimal policy of the corresponding MDP. Then the regret (with respect to the optimal policy corresponding to  $\phi^*$ ) of the BLB algorithm run with parameter  $\delta$ , for any horizon  $T$ , with probability higher than  $1 - \delta$  is bounded as follows*

$$\Delta(T) \leq c f(T) S \left( A J \log((J\delta)^{-1}) \log_2(T) \right)^{1/2} T^{2/3} + c' D S \left( A \log(\delta^{-1}) \log_2(T) T \right)^{1/2} + c(f, D), \quad (5)$$

for some numerical constants  $c, c'$  and  $c(f, D)$ . The parameter  $f(t)$  can be chosen to be any increasing function, for instance the choice  $f(t) := \log_2 t + 1$ , gives  $c(f, D) \leq 2^D$ .

The proof of this result is reported in Section 5.

**Remark.** Importantly, the algorithm considered here *does not* know in advance the diameter  $D$  of the true model, nor the time horizon  $T$ . Due to this lack of knowledge, it uses a guess  $f(t)$  (e.g.  $\log(t)$ ) on this diameter, which result in the additional regret term  $c(f, D)$  and the additional factor  $f(T)$ ; knowing  $D$  would enable to remove both of them, but this is a strong assumption. Choosing  $f(t) := \log_2 t + 1$  gives a bound which is of order  $T^{2/3}$  in  $T$  but is exponential in  $D$ ; taking  $f(t) := t^\varepsilon$  we get a bound of order  $T^{2/3+\varepsilon}$  in  $T$  but of polynomial order  $1/\varepsilon$  in  $D$ .

## 4 Discussion and outlook

**Intuition.** The main idea why this algorithm works is as follows. The “wrong” models are used during exploitation stages only as long as they are giving rewards that are higher than the rewards that could be obtained in the “true” model. All the models are explored sufficiently long so as to be able to estimate the optimal reward level in the true model, and to learn its policy. Thus, nothing has to be known about the “wrong” models. This is in stark contrast to the usual situation in mathematical statistics, where to be able to test a hypothesis about a model (e.g., that the data is generated by a certain model versus some alternative models), one has to make assumptions about alternative models. This has to be done in order to make sure that the Type II error is small (the power of the test is large): that this error is small has to be proven under the alternative. Here, although we are solving seemingly the same problem, the role of the Type II error is played by the rewards. As long as the rewards are high we do not care where the model we are using is correct or not. We only have to ensure that the true model passes the test.

**Assumptions.** A crucial assumption made in this work is that the “true” model  $\phi^*$  belongs to a known finite set. While passing from a finite to a countably infinite set appears rather straightforward, getting rid of the assumption that this set *contains* the true model seems more difficult. What one would want to obtain in this setting is sub-linear regret with respect to the performance of the optimal policy in the best model; this, however, seems difficult without additional assumptions on the probabilistic characteristics of the models. Another approach not discussed here would be to try to *build* a good state representation function, as what is suggested for instance in [5]. Yet another interesting generalization in this direction would be to consider uncountable (possibly parametric but general) sets of models. This, however, would necessarily require some heavy assumptions on the set of models.

**Regret.** The reader familiar with adversarial bandit literature will notice that our bound of order  $T^{2/3}$  is worse than  $T^{1/2}$  that usually appears in this context (see, for example [2]). The reason is that our notion of regret is different: in adversarial bandit literature, the regret is measured with respect to the best choice of the arm *for the given fixed history*. In contrast, we measure the regret with respect to the best policy (for knows the correct model and its parameters) that, in general, would obtain completely different (from what our algorithm would get) rewards and observations right from the beginning.

**Estimating the diameter?** As previously mentioned, a possibly large additive constant  $c(f, D)$  appears in the regret since we do not know a bound on the diameter of the MDP in the “true” model, and use  $\log T$  instead. Finding a way to properly address this problem by estimating online the diameter of the MDP is an interesting open question. Let us provide some intuition concerning this problem. First, we notice that, as reported in [6], when we compute an optimistic model based on the empirical rewards and transitions of the true model, the span of the corresponding optimistic value function  $sp(\widehat{V}^+)$  is always smaller than the diameter  $D$ . This span increases as we get more rewards and transitions samples, which gives a natural empirical lower bound on  $D$ . However, it seems quite difficult to compute a tight empirical upper bound on  $D$  (or  $sp(\widehat{V}^+)$ ). In [3], the authors derive a regret bound that scales with the span of the true value function  $sp(V^*)$ , which is also less than  $D$ , and can be significantly smaller in some cases. However, since we do not have the property that  $sp(\widehat{V}^+) \leq sp(V^*)$ , we need to introduce an explicit penalization in order to control the span of the computed optimistic models, and this requires assuming we know an upper bound  $B$  on  $sp(V^*)$  in order to guarantee a final regret bound scaling with  $B$ . Unfortunately this does not solve the estimation problem of  $D$ , which remains an open question.

## 5 Proof of Theorem 1

In this section, we now detail the proof of Theorem 1. The proof is stated in several parts. First we remind a general confidence bound for the UCRL2 algorithm in the true model. Then we decompose the regret into the sum of the regret in each stage  $i$ . After analyzing the contribution to the regret in stage  $i$ , we then gather all stages and tune the length of each stage and episode in order to get the final regret bound.

### 5.1 Upper and Lower confidence bounds

From the analysis of UCRL2 in [6], we have the property that with probability higher than  $1 - \delta'$ , the regret of UCRL2 when run for  $\tau$  consecutive many steps from time  $t_1$  in the true model  $\phi^*$  is upper bounded by

$$\rho^* - \frac{1}{\tau} \sum_{t=t_1}^{t_1+\tau-1} r_t \leq 34D |\mathcal{S}_{\phi^*}| \sqrt{\frac{A \log(\frac{\tau}{\delta'})}{\tau}}, \quad (6)$$

where  $D$  is the diameter of the MDP. What is interesting is that this diameter does not need to be known by the algorithm. Also by carefully looking at the proof of UCRL, it can be shown that the following bound is also valid with probability higher than  $1 - \delta'$ :

$$\frac{1}{\tau} \sum_{t=t_1}^{t_1+\tau-1} r_t - \rho^* \leq 34D |\mathcal{S}_{\phi^*}| \sqrt{\frac{A \log(\frac{\tau}{\delta'})}{\tau}}.$$

We now define the following quantity, for every model  $\phi$ , episode length  $\tau$  and  $\delta' \in (0, 1)$

$$B_D(\tau, \phi, \delta') \stackrel{\text{def}}{=} 34D |\mathcal{S}_{\phi}| \sqrt{\frac{A \log(\frac{\tau}{\delta'})}{\tau}}. \quad (7)$$

### 5.2 Regret of stage $i$

In this section we analyze the regret of the stage  $i$ , which we denote  $\Delta_i$ . Note that since each stage  $i \leq I$  is of length  $\tau_i = 2^i$  except the last one  $I$  that may stop before, we have

$$\Delta(T) = \sum_{i=1}^{I(T)} \Delta_i, \quad (8)$$

where  $I(T) = \lfloor \log_2(T+1) \rfloor$ . We further decompose  $\Delta_i = \Delta_{1,i} + \Delta_{i,2}$  into the regret corresponding to the exploration stage  $\Delta_{1,i}$  and the regret corresponding to the exploitation stage  $\Delta_{i,2}$ .

Recall that  $\tau_{i,1}$  is the total length of the exploration stage  $i$  and  $\tau_{i,2}$  is the total length of the exploitation stage  $i$ . Then for each model  $\phi$ , we write  $\tau_{i,1,J} \stackrel{\text{def}}{=} \frac{\tau_{i,1}}{J}$  the number of consecutive steps during which the UCRL2 algorithm is run with model  $\phi$  in the exploration stage  $i$ , and  $\tau_{i,2}(\phi)$  the number of consecutive steps during which the UCRL2 algorithm is run with model  $\phi$  in the exploitation stage  $i$ .

**Good and Bad models.** Let us now introduce the two following sets of models, defined after the end of the exploration stage, i.e. at time  $t_i$ .

$$\begin{aligned}\mathcal{G}_i &\stackrel{\text{def}}{=} \{\phi \in \Phi ; \widehat{\mu}_{i,1}(\phi) - 2B(i, \phi, \delta) \geq \widehat{\mu}_{i,1}(\phi^*) - 2B(i, \phi^*, \delta)\} \setminus \{\phi^*\}, \\ \mathcal{B}_i &\stackrel{\text{def}}{=} \{\phi \in \Phi ; \widehat{\mu}_{i,1}(\phi) - 2B(i, \phi, \delta) < \widehat{\mu}_{i,1}(\phi^*) - 2B(i, \phi^*, \delta)\}.\end{aligned}$$

With this definition, we have the decomposition  $\Phi = \mathcal{G}_i \cup \{\phi^*\} \cup \mathcal{B}_i$ .

### 5.2.1 Regret in the exploration phase

Since in the exploration stage  $i$  each model  $\phi$  is run for  $\tau_{i,1,J}$  many steps, the regret for each model  $\phi \neq \phi^*$  is bounded by  $\tau_{i,1,J}\rho^*$ . Now the regret for the true model is  $\tau_{i,1,J}(\rho^* - \widehat{\mu}_1(\phi^*))$ , thus the total contribution to the regret in the exploration stage  $i$  is upper-bounded by

$$\Delta_{i,1} \leq \tau_{i,1,J}(\rho^* - \widehat{\mu}_1(\phi^*)) + (J-1)\tau_{i,1,J}\rho^*. \quad (9)$$

### 5.2.2 Regret in the exploitation phase

By definition, all models in  $\mathcal{G}_i \cup \{\phi^*\}$  are selected before any model in  $\mathcal{B}_i$  is selected.

**The good models.** Let us consider some  $\phi \in \mathcal{G}_i$  and an event  $\Omega_i$  under which the exploitation phase does not reset. The test (equation (4)) starts after  $\tau_{i,1,J}$ , thus, since there is not reset, either  $\tau_{i,2}(\phi) = \tau_{i,1,J}$  in which case the contribution to the regret is bounded by  $\tau_{i,1,J}\rho^*$ , or  $\tau_{i,2}(\phi) > \tau_{i,1,J}$ , in which case the regret during the  $(\tau_{i,2}(\phi) - 1)$  steps (where the test was successful) is bounded by

$$\begin{aligned}(\tau_{i,2}(\phi) - 1)(\rho^* - \widehat{\mu}_{i,2,\tau_{i,2}(\phi)-1}(\phi)) &\leq (\tau_{i,2}(\phi) - 1)(\rho^* - \widehat{\mu}_{i,1}(\phi) + 2B(i, \phi, \delta)) \\ &\leq (\tau_{i,2}(\phi) - 1)(\rho^* - \widehat{\mu}_{i,1}(\phi^*) + 2B(i, \phi^*, \delta)),\end{aligned}$$

and now since in the last step  $\phi$  fails to pass the test, this adds a contribution to the regret at most  $\rho^*$ .

We deduce that the total contribution to the regret of all the models  $\phi \in \mathcal{G}_i$  in the exploitation stages on the event  $\Omega_i$  is bounded by

$$\Delta_{i,2}(\mathcal{G}_i) \leq \sum_{\phi \in \mathcal{G}_i} \max\{\tau_{i,1,J}\rho^*, (\tau_{i,2}(\phi) - 1)(\rho^* - \widehat{\mu}_{i,1}(\phi^*) + 2B(i, \phi^*, \delta)) + \rho^*\}. \quad (10)$$

**The true model.** First, let us note that since the total regret of the true model during the exploitation step  $i$  is given by

$$\tau_{i,2}(\phi^*)(\rho^* - \widehat{\mu}_{i,2,t}(\phi^*)),$$

then the total regret of the exploration and exploitation stages in episode  $i$  on  $\Omega_i$  is bounded by

$$\begin{aligned}\Delta_i &\leq \tau_{i,1,J}(\rho^* - \widehat{\mu}_1(\phi^*)) + \tau_{i,1,J}(J-1)\rho^* + \tau_{i,2}(\phi^*)(\rho^* - \widehat{\mu}_{i,2,t_i+\tau_{i,2}}(\phi^*)) + \\ &\quad \sum_{\phi \in \mathcal{G}_i} \max\{\tau_{i,1,J}\rho^*, (\tau_{i,2}(\phi) - 1)(\rho^* - \widehat{\mu}_{i,1}(\phi^*) + 2B(i, \phi^*, \delta)) + \rho^*\} + \sum_{\phi \in \mathcal{B}_i} \tau_{i,2}(\phi)\rho^*.\end{aligned}$$

Now from the analysis provided in [6] we know that when we run the UCRL2 with the true model  $\phi^*$  with parameter  $\delta_i(\delta)$ , then there exists an event  $\Omega_{1,i}$  of probability at least  $1 - \delta_i(\delta)$  such that on this event

$$\rho^* - \widehat{\mu}_{i,1}(\phi^*) \leq B_D(\tau_{i,1,J}, \phi^*, \delta_i(\delta)),$$

and similarly there exists an event  $\Omega_{2,i}$  of probability at least  $1 - \delta_i(\delta)$ , such that on this event

$$\rho^* - \widehat{\mu}_{i,2,t}(\phi^*) \leq B_D(\tau_{i,2}(\phi^*), \phi^*, \delta_1(\delta)).$$

Now we show that, with high probability, the true model  $\phi^*$  passes all the tests (equation (4)) until the end of the episode  $i$ , and thus equivalently, with high probability no model  $\phi \in \mathcal{B}_i$  is selected, so that  $\sum_{\phi \in \mathcal{B}_i} \tau_{i,2}(\phi) = 0$ .

For the true model, after  $\tau(\phi^*, t) \geq \tau_{i,1,J}$ , there remains at most  $(\tau_{i,2} - \tau_{i,1,J} + 1)$  possible timesteps where we do the test for the true model  $\phi^*$ . For each test we need to control  $\mu_{i,2,t}(\phi^*)$ , and the event corresponding to  $\widehat{\mu}_{i,1}(\phi^*)$  is shared by all the tests. Thus we deduce that with probability higher than  $1 - (\tau_{i,2} - \tau_{i,1,J} + 2)\delta_i(\delta)$  we have simultaneously on all time step until the end of exploitation phase of stage  $i$ ,

$$\begin{aligned}\widehat{\mu}_{i,2,t}(\phi^*) - \widehat{\mu}_{i,1}(\phi^*) &= \widehat{\mu}_{i,2,t}(\phi^*) - \rho^* + \rho^* - \widehat{\mu}_{i,1}(\phi^*) \\ &\geq -B_D(\tau(\phi^*, t), \phi^*, \delta_i(\delta)) - B_D(\tau_{i,1,J}, \phi^*, \delta_i(\delta)) \\ &\geq -2B_D(\tau_{i,1,J}, \phi^*, \delta_i(\delta)).\end{aligned}$$

Now provided that  $f(t_i) \geq D$ , then  $B_D(\tau_{i,1,J}, \phi^*, \delta_i(\delta)) \leq B(i, \phi^*, \delta)$ , thus the true model passes all tests until the end of the exploitation part of stage  $i$  on an event  $\Omega_{3,i}$  of probability higher than  $1 - (\tau_{i,2} - \tau_{i,1,J} + 2)\delta_i(\delta)$ . Since there is no reset, we can choose  $\Omega_i \stackrel{\text{def}}{=} \Omega_{3,i}$ . Note that on this event, we thus have  $\sum_{\phi \in \mathcal{B}_i} \tau_{i,2}(\phi) = 0$ .

By using a union bound over the events  $\Omega_{1,i}, \Omega_{2,i}$  and  $\Omega_{3,i}$ , then we deduce that with probability higher than  $1 - (\tau_{i,2} - \tau_{i,1,J} + 4)\delta_i(\delta)$ ,

$$\begin{aligned} \Delta_i &\leq \tau_{i,1,J} B_D(\tau_{i,1,J}, \phi^*, \delta_i(\delta)) + [\tau_{i,1,J}(J-1) + |\mathcal{G}_i|] \rho^* + \tau_{i,2}(\phi^*) B_D(\tau_{i,2}(\phi^*), \phi^*, \delta_i(\delta)) \\ &\quad + \sum_{\phi \in \mathcal{G}_i} \max\{(\tau_{i,1,J} - 1)\rho^*, (\tau_{i,2}(\phi) - 1)(B_D(\tau_{i,1,J}, \phi^*, \delta_i(\delta)) + 2B(i, \phi^*, \delta))\}. \end{aligned}$$

Now using again the fact that  $f(t_i) \geq D$ , and after some simplifications, we deduce that

$$\begin{aligned} \Delta_i &\leq \tau_{i,1,J} B_D(\tau_{i,1,J}, \phi^*, \delta_i(\delta)) + \tau_{i,2}(\phi^*) B_D(\tau_{i,2}(\phi^*), \phi^*, \delta_i(\delta)) \\ &\quad + \sum_{\phi \in \mathcal{G}_i} (\tau_{i,2}(\phi) - 1) 3B(i, \phi^*, \delta) + \tau_{i,1,J}(J + |\mathcal{G}_i| - 1)\rho^*. \end{aligned}$$

Finally, we use the fact that  $\tau B_D(\tau, \phi^*, \delta_i(\delta))$  is increasing with  $\tau$  to deduce the following rough bound that holds with probability higher than  $1 - (\tau_{i,2} - \tau_{i,1,J} + 4)\delta_i(\delta)$

$$\Delta_i \leq \tau_{i,2} B(i, \phi^*, \delta) + \tau_{i,2} B_D(\tau_{i,2}, \phi^*, \delta_i(\delta)) + 2J\tau_{i,1,J}\rho^*,$$

where we used the fact that  $\tau_{i,2} = \tau_{i,2}(\phi^*) + \sum_{\phi \in \mathcal{G}} \tau_{i,2}(\phi)$ .

### 5.3 Tuning the parameters of each stage.

We now conclude by tuning the parameters of each stage, i.e. the probabilities  $\delta_i(\delta)$  and the length  $\tau_i, \tau_{i,1}$  and  $\tau_{i,2}$ . The total length of stage  $i$  is by definition

$$\tau_i = \tau_{i,1} + \tau_{i,2} = \tau_{i,1,J}J + \tau_{i,2},$$

where  $\tau_i = 2^i$ . So we set  $\tau_{i,1} \stackrel{\text{def}}{=} \tau_i^{2/3}$  and then we have  $\tau_{i,2} \stackrel{\text{def}}{=} \tau_i - \tau_i^{2/3}$  and  $\tau_{i,1,J} = \frac{\tau_i^{2/3}}{J}$ . Now using these values and the definition of the bound  $B(i, \phi^*, \delta)$ , and  $B_D(\tau_{i,2}, \phi^*, \delta_i(\delta))$ , we deduce with probability higher than  $1 - (\tau_{i,2} - \tau_{i,1,J} + 4)\delta_i(\delta)$  the following upper bound

$$\Delta_i \leq 34f(t_i)S \sqrt{AJ \log\left(\frac{\tau_i^{2/3}}{J\delta_i(\delta)}\right) \tau_i^{2/3}} + 34DS \sqrt{A \log\left(\frac{\tau_i}{\delta_i(\delta)}\right) \tau_i} + 2\tau_i^{2/3}\rho^*,$$

with  $t_i = 2^i - 1 + 2^{2i/3}$  and where we used the fact that  $\left(\frac{J}{\tau_i^{2/3}}\right)^{1/2} \tau_{i,2} \leq \sqrt{J}\tau_i^{2/3}$ .

We now define  $\delta_i(\delta)$  such that  $\delta_i(\delta) \stackrel{\text{def}}{=} (2^i - (J^{-1} + 1)2^{2i/3} + 4)^{-1}2^{-i+1}\delta$ .

Since for the stages  $i \in \mathcal{I}_0 \stackrel{\text{def}}{=} \{i \geq 1; f(t_i) < D\}$ , the regret is bounded by  $\Delta_i \leq \tau_i \rho^*$ , then the total cumulative regret of the algorithm is bounded with probability higher than  $1 - \delta$  (using the definition of the  $\delta_i(\delta)$ ) by

$$\Delta(T) \leq \sum_{i \notin \mathcal{I}_0} [34f(t_i)S \sqrt{JA \log\left(\frac{2^{8i/3}}{J\delta}\right) + 2} 2^{2i/3} + 34DS \sqrt{A \log\left(\frac{2^{3i}}{\delta}\right) 2^i} + \sum_{i \in \mathcal{I}_0} 2^i \rho^*].$$

where  $t_i = 2^i - 1 + 2^{2i/3} \leq T$ .

We conclude by using the fact that since  $I(T) \leq \log_2(T+1)$ , then with probability higher than  $1 - \delta$ , the following bound on the regret holds

$$\Delta(T) \leq cf(T)S \left(AJ \log(J\delta)^{-1} \log_2(T)\right)^{1/2} T^{2/3} + c'DS \left(A \log(\delta^{-1}) \log_2(T)T\right)^{1/2} + c(f, D).$$

for some constant  $c, c'$ , and where  $c(f, D) = \sum_{i \in \mathcal{I}_0} 2^i \rho^*$ . Now for the special choice when  $f(T) \stackrel{\text{def}}{=} \log_2(T+1)$ , then  $i \in \mathcal{I}_0$  means  $2^i + 2^{2i/3} < 2^D + 2$ , thus we must have  $i < D$ , and thus  $c(f, d) \leq 2^D$ .

## Acknowledgements

This research was partially supported by the French Ministry of Higher Education and Research, Nord- Pas-de-Calais Regional Council and FEDER through CPER 2007-2013, ANR projects EXPLO-RA (ANR-08-COSI-004) and Lampada (ANR-09-EMER-007), by the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement 231495 (project ComplACS), and by Pascal-2.

## References

- [1] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.
- [2] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 322–331, oct 1995.
- [3] Peter L. Bartlett and Ambuj Tewari. REGAL: a regularization based algorithm for reinforcement learning in weakly communicating mdps. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI*, pages 35–42, Arlington, Virginia, United States, 2009. AUAI Press.
- [4] Ronen I. Brafman and Moshe Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, March 2003.
- [5] Marcus Hutter. Feature reinforcement learning: Part I: Unstructured MDPs. *Journal of Artificial General Intelligence*, 1:3–24, 2009.
- [6] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 99:1563–1600, August 2010.
- [7] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49:209–232, November 2002.
- [8] Tze L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.
- [9] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematics Society*, 58:527–535, 1952.
- [10] Daniil Ryabko and Marcus Hutter. On the possibility of learning in reactive environments with arbitrary dependence. *Theoretical Computer Science*, 405:274–284, October 2008.
- [11] Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. PAC model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning, ICML*, pages 881–888, New York, NY, USA, 2006. ACM.
- [12] Ambuj Tewari and Peter L. Bartlett. Optimistic linear programming gives logarithmic regret for irreducible mdps. In *Proceedings of Neural Information Processing Systems Conference (NIPS)*, 2007.