Navigating to the Best Policy in Markov Decision Processes

Anonymous Author(s)
Affiliation
Address
email

Abstract

We investigate the classical active pure exploration problem in Markov Decision Processes, where the agent sequentially selects actions and, from the resulting system trajectory, aims at identifying the best policy as fast as possible. We propose an information-theoretic lower bound on the average number of steps required before a correct answer can be given with probability at least $1-\delta$. This lower bound involves a non-convex optimization problem, for which we propose a convex relaxation. We further provide an algorithm whose sample complexity matches the relaxed lower bound up to a factor 2. This algorithm addresses general communicating MDPs; we propose a variant with reduced exploration rate (and hence faster convergence) under an additional ergodicity assumption. This work extends previous results relative to the *generative setting* [MP20], where the agent could at each step observe the random outcome of any (state, action) pair. In contrast, we show here how to deal with the *navigation constraints* and comment on the gap between the generative and online settings. Our analysis relies on an ergodic theorem for non-homogeneous Markov chains which we consider of wide interest in the analysis of Markov Decision Processes.

1 Introduction and related work

Somewhat surprisingly, learning in a Markov Decision Process is most often considered under the performance criteria of *consistency* or *regret minimization* (see e.g. [SB18, Sze10, LS20] and references therein). Regret minimization (see e.g. [AJO09, FCG10]) is particularly relevant when the rewards accumulated during the learning phase are important. This is however not always the case: for example, when learning a game (whether Go, chess, Atari, or whatever), winning or losing during the learning phase does not really matter. One may intuitively think that sometimes getting into difficulty on purpose so as to observe unheard situations can significantly accelerate the learning process. Another example is the training of robot prototypes in the factory: a reasonably good policy is first searched, regardless of the losses incurred, that can serve as an initialization for a second phase regret-minimization mode that starts when the robot is deployed. It is hence also of great practical importance to study the sample complexity of learning, and to work on strategies that might improve, in this perspective, on regret minimizing algorithms.

Pure exploration problems in MDPs have been mostly investigated in the so-called PAC-MDP framework [Kak03], where the sample complexity is defined as the deterministic number of samples needed to output an ε-optimal policy with probability at least $1-\delta$.[1] Existing results concern the minimax sample complexity [KS99], [Kak03], [EDMM06], [AMK13], [SWW+18], [AKY20], [LWC+20a], [LWC+20b] [AHKS20], [ZKOB20], [FYAY21], [ZCA21], [Din21]. The minimax

[1] Some works define the sample complexity of a PAC-MDP algorithm as the number of time steps where the algorithm acts ε-sub-optimaly, see discussion in [DMKV21] and references therein.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

framework may be overly pessimistic by accounting for the worst possible MDP, whereas an algorithm with instance-dependent guarantees can adapt its exploration procedure to the hardness of the MDP instance that it faces. It is worth noting that a large body of literature focuses on the so-called *generative model*, where in each step, the algorithm may sample a transition and a reward from any given (state, action) pair [KS98], [EDMM06], [AMK13],[DTC13], [Wan17], [SWW+18], [AKY20], [LWC+20a], [LCC+21].

In this work, we are interested in the best policy identification (BPI) problem for infinite-horizon discounted MDPs. This framework was introduced by [Fie94] under the name of *PAC-RL*. In BPI the algorithm halts at a *random time step*, called the sample complexity. This random time is determined by a *stopping rule* that guarantees that the algorithm is able to return an ε-optimal policy with probability at least $1-\delta$. BPI has been investigated recently by a number of works [ZKB19],[MP20] with instance-dependent sample complexity and by [KMDD+21],[MDJ+20] in the minimax framework. However they typically either make simplifying assumptions such as a generative model [ZKB19, MP20], the possibility to reset the system to some state [Fie94], or restrict their attention to episodic MDPs [KMDD+21, MDJ+20]. In practice however, the samples gathered rather correspond to a single trajectory of the system that we wish to control. We study the full online setting where observations are only obtained by navigating in the MDP, that is by sequential choices of actions following the transitions of the MDP. We hope to extend the information-theoretical analysis for best-arm identification in stochastic Multi-Armed Bandit problems [GK16, Rus16] to the problem of best policy identification in MDPs. [MP20] recently proposes an information-theoretical complexity analysis for MDPs in the case of a generative model. The online setting induces new *navigation constraints*: we discuss in this paper how to include them into the analysis, and what is their impact on the sample complexity. Our work builds on to [TL19], where the problem of actively estimating the rewards in an ergodic MDPs with known transitions is considered. Similar to our setting, there the authors seek an optimal allocation, minimizing a weighted sum of variances of mean reward estimation, subject to navigation constraints. In contrast, we here take a more direct approach to the full learning problem by looking for a strategy that minimizes the sample complexity of finding the optimal policy. Our contributions are mostly theoretical, but our proposals aim to open the way to more sample-efficient methods for learning in MDPs.

Our main contributions are the following. After giving some notation and definitions in Section 2, we first derive a lower bound on the time required by any algorithm navigating the MDP until it is able to identify the best policy with probability at least $1-\delta$: it scales as $\log(1/\delta)$ multiplied by the solution of a non-convex optimization problem (Proposition 3 in Section 3). We leverage a tractable upper-bound of the optimization-problem towards the design of an algorithm in Section 4, whose sample complexity matches the relaxed lower bound up to a factor of 2. While this algorithm addresses general communicating MDPs, we propose a variant with reduced exploration rate (and hence faster convergence) under an additional ergodicity assumption. The analysis of the algorithm, sketched in Section 5, relies (among other ingredients) on an ergodic theorem for non-homogeneous Markov chains of independent interest (Proposition 13). Most technical results and proofs are given in the appendix.

2 Preliminaries and Notations

We consider infinite horizon MDPs with discount and finite state and action spaces $\mathcal{S}$ and $\mathcal{A}$. Such an MDP $\mathcal{M}$ is defined through its transition and reward distributions $p_{\mathcal{M}}(\cdot|s,a)$ and $q_{\mathcal{M}}(\cdot|s,a)$ for all (s,a). For simplicity, $q_{\mathcal{M}}(\cdot|s,a)$ will denote the density of the reward distribution w.r.t. a positive measure λ with support included in $[0,1]$. We assume that for all MDP $\mathcal{M}$, this density is strictly positive. The average reward in (s,a) is $r_{\mathcal{M}}(s,a)$. $\gamma \in ([0,1)$ is a discounting factor. We look for an optimal control policy $\pi^\star : \mathcal{S} \to \mathcal{A}$ maximizing the long-term discounted reward $\mathbb{E}_{\mathcal{M},\pi^\star}[\sum_{t=0}^{\infty} \gamma^t r_{\mathcal{M}}(s_t,a_t)]$, where $\mathbb{E}_{\mathcal{M},\pi^\star}$ denotes the expectation when $\pi^\star$ is used. Classically, we denote by $V^\pi_{\mathcal{M}}$ the state value function of the policy π, by $V^\star_{\mathcal{M}}$ the value function and by $Q^\star_{\mathcal{M}}$ the Q-function of $\mathcal{M}$. Denote by $\Pi^\star(\mathcal{M})$ the set of optimal policies for $\mathcal{M}$. The sub-optimality gap of action a in state s is defined as $\Delta_{sa} = V^\star_{\mathcal{M}}(s) - Q^\star_{\mathcal{M}}(s,a)$. Let $\Delta_{\min} = \min_{s,a\neq\pi^\star(s)} \Delta_{s,a}$ and $\mathrm{sp}(V^\star_{\mathcal{M}}) = \max_{s,s'} |V^\star_{\mathcal{M}}(s) - V^\star_{\mathcal{M}}(s')|$ the span of $V^\star_{\mathcal{M}}$. Finally we introduce the variance of the value function in (s,a) as $\mathrm{Var}_{p_{\mathcal{M}}(s,a)}[V^\star_{\mathcal{M}}] = \mathrm{Var}_{s'\sim p_{\mathcal{M}}(.|s,a)}[V^\star_{\mathcal{M}}(s')]$.

Active pure exploration with fixed confidence. When $\mathcal{M}$ is unknown, we wish to devise a learning algorithm $\mathbb{A}$ identifying, from a single trajectory, an optimal policy as quickly as possible with some

given level of confidence. Formally, such an algorithm consists of (i) a sampling rule, selecting in each round t in an adaptive manner the action a_t to be applied; a_t depends on past observations, it is $\mathcal{F}_t^{\mathbb{A}}$-measurable where $\mathcal{F}_t^{\mathbb{A}} = \sigma(s_0, a_0, R_0 \ldots, s_{(t-1)}, a_{(t-1)}, R_{(t-1)}, s_t)$; (ii) A stopping rule τ_δ, a stopping time w.r.t. the filtration $(\mathcal{F}_t^{\mathbb{A}})_{t\geq 0}$, deciding when to stop observing; (iii) a decision rule returning $\hat{\pi}^\star_{\tau_\delta}$ an estimated optimal policy.

An algorithm $\mathbb{A}$ is δ-PC over some set $\mathbb{M}$ of MDPs, if for any $\mathcal{M} \in \mathbb{M}$, it returns (in finite time) an optimal policy with probability at least $1-\delta$. In this paper, we devise a δ-PC algorithm $\mathbb{A}$ with minimal sample complexity $\mathbb{E}_{\mathcal{M},\mathbb{A}}[\tau_\delta]$. We make the following assumption.

Assumption 1. *We consider the set $\mathbb{M}$ of communicating MDPs with a unique optimal policy.*

We justify the above assumption as follows. (i) We restrict our attention to the case where $\mathcal{M}$ is communicating, for otherwise, if it is Multichain and there would be a non-zero probability that the algorithm enters a subclass of states from which there is no possible comeback. In this case it becomes impossible to identify the *global* optimal policy[2]. (ii) About the uniqueness of the optimal policy, treating the case of MDPs with multiple optimal policies requires the use of more involved Overlapping Hypothesis tests, which is already challenging in multi-arm bandits (MDPs with a single state) [GK19]. We will analyze how to remove this assumption in future work.

Notations. $\mathcal{Z} \triangleq \mathcal{S} \times \mathcal{A}$ is the state-action space. $\Sigma \triangleq \{\omega \in \mathbb{R}_+^{S\times A} : \sum_{s,a} \omega_{sa} = 1\}$, the simplex of dimension $SA-1$. With some abuse of notation, for a vector $\omega \in \Sigma$ we will denote $\omega_s \triangleq \sum_{a\in\mathcal{A}} \omega_{sa}$. For a matrix $A \in \mathbb{R}^{n\times m}$, the infinity norm is defined as $\|A\|_\infty \triangleq \max_{1\leq i\leq n} \sum_{j=1}^m |a_{i,j}|$. The KL divergence between two probability distributions P and Q on some discrete space $\mathcal{S}$ is defined as: $KL(P\|Q) \triangleq \sum_{s\in\mathcal{S}} P(s)\log(\frac{P(s)}{Q(s)})$. For Bernoulli distributions of respective means p and q, the KL divergence is denoted by $\mathrm{kl}(p,q)$. For distributions over $[0,1]$ defined through their densities p and q w.r.t. some positive measure λ, the KL divergence is: $KL(p\|q) \triangleq \int_{-\infty}^{\infty} p(x)\log\left(\frac{p(x)}{q(x)}\right)\lambda(dx)$. For two MDPs $\mathcal{M}, \mathcal{M}' \in \mathbb{M}$, we say that $\mathcal{M} \ll \mathcal{M}'$ if for all (s,a), $p_{\mathcal{M}}(\cdot|s,a) \ll p_{\mathcal{M}'}(\cdot|s,a)$. In that case, for any state (action pair) (s,a), we define the KL divergence of the distributions of the one-step observations under $\mathcal{M}$ and $\mathcal{M}'$ when starting at (s,a) as $\mathrm{KL}_{\mathcal{M}|\mathcal{M}'}(s,a) \triangleq KL(p_{\mathcal{M}}(\cdot|s,a)\|p_{\mathcal{M}'}(\cdot|s,a)) + KL(q_{\mathcal{M}}(\cdot|s,a)\|q_{\mathcal{M}'}(\cdot|s,a))$.

3 Sample Complexity Lower Bounds

In this section, we first derive a lower bound on the expected sample complexity satisfied by any δ-PC algorithm. The lower bound is obtained as the solution of a non-convex optimization problem, as in the case where the learner has access to a generative model [MP20]. The problem has however additional constraints, referred to as the *navigation* constraints, due the fact that the learner has access to a single system trajectory. We then propose an upper bound of the sample complexity lower bound. This new bound corresponds to an allocation satisfying the navigation constraint, and is simply given as the solution of a convex optimization problem.

The expected sample complexity of an algorithm $\mathbb{A}$ is $\mathbb{E}_{\mathcal{M},\mathbb{A}}[\tau_\delta] = \sum_{s,a} \mathbb{E}_{\mathcal{M},\mathbb{A}}[N_{sa}(\tau_\delta)]$, where $N_{sa}(t)$ denotes the number of times the (state, action) pair (s,a) has been visited up to the end of round t. We also introduce $N_s(t) = \sum_a N_{sa}(t)$. Lower bounds on the sample complexity are derived by identifying constraints that the various $N_{sa}(\tau_\delta)$'s need to satisfy so as to get a δ-PC algorithm. We distinguish here two kinds of constraints: (i) information constraints expressing the need for the algorithm to distinguish between different possible MDPs; (ii) Navigation constraints obtained by considering the system dynamics.

Information constraints. We first identifying constraints on $\mathbb{E}_{\mathcal{M},\mathbb{A}}[N_{sa}(\tau_\delta)]$, so that the algorithm can learn the optimal policy with probability at least $1-\delta$. These constraints are referred to as information constraints and are the same as those derived when the learner has access to a generative model [MP20]. Let $\mathrm{Alt}(\mathcal{M})$ be the set of MDPs such that $\mathcal{M} \ll \mathcal{M}'$ and $\Pi^\star(\mathcal{M}) \cap \Pi^\star(\mathcal{M}') = \emptyset$. The information constraints are obtained by change-of-measure arguments as in the bandit literature [LR85, KCG16]:

[2] Unless we modify the objective to finding the optimal policy in this subchain.

Lemma 1. *([MP20]) For any δ-PC algorithm $\mathbb{A}$, and for any $\mathcal{M}' \in \mathrm{Alt}(\mathcal{M})$, we have:*

$$\sum_{s,a} \mathbb{E}_{\mathcal{M},\mathbb{A}}[N_{sa}(\tau_\delta)] KL_{\mathcal{M}|\mathcal{M}'}(s,a) \geq \mathrm{kl}(\delta, 1-\delta). \tag{1}$$

Navigation constraints. Observations come from a single (but controlled) system trajectory which imposes additional constraints on the $N_{sa}(\tau_\delta)$'s. These are derived by just writing the Chapman-Kolmogorov equations of the controlled Markov chain (refer to the appendix for a proof).

Lemma 2. *For any algorithm $\mathbb{A}$, and for any state s, we have:*

$$\left| \mathbb{E}_{\mathcal{M},\mathbb{A}}[N_s(\tau_\delta)] - \sum_{y,a} p_{\mathcal{M}}(s|y,a) \mathbb{E}_{\mathcal{M},\mathbb{A}}[N_{ya}(\tau_\delta)] \right| \leq 1. \tag{2}$$

Putting all constraints together, we obtain the following sample complexity lower bound.

Proposition 3. *The expected sample complexity $\mathbb{E}_{\mathcal{M},\mathbb{A}}[\tau_\delta]$ of any δ-PC algorithm $\mathbb{A}$ is larger than the value of the following optimization problem:*

$$\begin{aligned} &\inf_{n \geq 0} \sum_{s,a} n_{sa} \\ &\textit{s.t. } \forall s, \ \Big| \sum_a n_{sa} - \sum_{y,a} p_{\mathcal{M}}(s|y,a) n_{ya} \Big| \leq 1, \\ &\qquad \forall \mathcal{M}' \in \mathrm{Alt}(\mathcal{M}), \ \sum_{s,a} n_{sa} KL_{\mathcal{M}|\mathcal{M}'}(s,a) \geq \mathrm{kl}(\delta, 1-\delta). \end{aligned} \tag{3}$$

In (3), n_{sa} is interpreted as the expected number of times (s,a) is visited before the stopping time. Note that the above proposition provides a lower bound for any value of δ. We can further simplify this bound when restricting our attention to asymptotic regimes where δ goes to 0. In that case, the navigation constraints (2) can be replaced to $\sum_a n_{sa} = \sum_{y,a} p_{\mathcal{M}}(s|y,a) n_{ya}$ (by just renormalizing n with $n/\mathrm{kl}(\delta, 1-\delta)$, and letting $\delta \to 0$). For small δ regimes, we can hence rewrite the lower bound as follows. Define the set of navigation-constrained allocation vectors:

$$\Omega(\mathcal{M}) \triangleq \big\{ \omega \in \Sigma : \ \forall s \in \mathcal{S}, \ \omega_s = \sum_{s',a'} p_{\mathcal{M}}(s|s',a') \omega_{s'a'} \big\}.$$

ω_{sa} will be then interpreted as the proportion of times the algorithm visits (s,a).

Proposition 4. *The expected sample complexity $\mathbb{E}_{\mathcal{M},\mathbb{A}}[\tau_\delta]$ of any δ-PC algorithm $\mathbb{A}$ satisfies:*

$$\liminf_{\delta \to 0} \frac{\mathbb{E}_{\mathcal{M},\mathbb{A}}[\tau]}{\mathrm{kl}(\delta, 1-\delta)} \geq T_o(\mathcal{M}), \quad \textit{where} \quad T_o(\mathcal{M})^{-1} = \sup_{\omega \in \Omega(\mathcal{M})} T(\mathcal{M}, \omega)^{-1} \tag{4}$$

and

$$T(\mathcal{M}, \omega)^{-1} = \inf_{\mathcal{M}' \in \mathrm{Alt}(\mathcal{M})} \sum_{s,a} \omega_{sa} KL_{\mathcal{M}|\mathcal{M}'}(s,a). \tag{5}$$

As shown in [MP20], even without accounting for the navigation constraints, computing the characteristic time $T_o(\mathcal{M})$ and in particular, the optimal allocation leading to it is not easy. Indeed, the sub-problem corresponding to computing $T(\mathcal{M}, \omega)^{-1}$ is non-convex. [MP20] provides a tractable upper bound $U(\mathcal{M}, \omega)$ of $T(\mathcal{M}, \omega)$ which we recall here:

Lemma 5. *(Theorem 1, [MP20]) For all vectors $\omega \in \Sigma$, $T(\mathcal{M}, \omega) \leq U(\mathcal{M}, \omega)$, where*[3]

$$U(\mathcal{M}, \omega) = \max_{(s,a): a \neq \pi^*(s)} \frac{H_{sa}}{\omega_{sa}} + \frac{H^\star}{S \min_s \omega_{s,\pi^\star(s)}}, \tag{6}$$

$$\textit{and} \begin{cases} H_{sa} = \dfrac{2}{\Delta_{sa}^2} + \max\left(\dfrac{16 \mathrm{Var}_{p(s,a)}[V_{\mathcal{M}}^\star]}{\Delta_{sa}^2}, \dfrac{6 \, \mathrm{sp}(V_{\mathcal{M}}^\star)^{4/3}}{\Delta_{sa}^{4/3}} \right), \\ H^\star = \dfrac{2S}{[\Delta_{\min}(1-\gamma)]^2} + \mathcal{O}\left(\dfrac{S}{\Delta_{\min}^2 (1-\gamma)^3} \right). \end{cases} \tag{7}$$

[3] The exact definition of $H^\star$ is given in the appendix.

Using $U(\mathcal{M}, \omega)$, we obtain the following upper bound on the characteristic time (4):

$$T_o(\mathcal{M}) \leq U_o(\mathcal{M}) \triangleq \inf_{\omega \in \Omega(\mathcal{M})} U(\mathcal{M}, \omega). \tag{8}$$

In the same spirit as in [ZKB19, MP20], we can interpret H_{sa} (resp. $H^\star$) in the upper bound as proxys for the sample complexity of learning that the pair (s, a) is suboptimal (resp. of confirming that pairs $(s', \pi^\star(s'))$ are optimal). Hence, the optimization problem (8) states that, subject to the constraint of following a single trajectory, a sample-efficient BPI algorithm must try to visit the state-action pairs of $\mathcal{M}$ proportionally to the complexity of learning their status (i.e., whether they are optimal or not). While $U_o(\mathcal{M})$ is still an instance-dependent quantity, it is not straightforward to grasp how it scales with the functionals of $\mathcal{M}$. The following result gives a (loose) bound on its scaling.

Corollary 1. *Let π be an arbitrary stationary policy, and let $\mu = (\mu_{sa})_{s,a}$ its state-action occupancy vector where $\mu_{sa} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{P}_{\mathcal{M},\pi}(s_t = s, a_t = a)$. Define $\mu_{\min} = \min_{s,a} \mu_{sa}$. Then $\mu \in \Omega(\mathcal{M})$. As a result $U_o(\mathcal{M}) = \mathcal{O}\big(\frac{1}{\mu_{\min}\Delta_{\min}^2(1-\gamma)^3}\big)$.*

The scaling obtained above is similar to that obtained in [LWC+20b] for the sample complexity of identifying an ε-estimate of the Q-function via Asynchronous variance-reduced Q-Learning. There, the scaling is $\mathcal{O}\big(1/(\mu_{\min}\varepsilon^2(1-\gamma)^3)\big)$. To identify the optimal policy, it is sufficient (and in the worst case necessary) to set $\varepsilon < \Delta_{\min}/2$ and return the greedy policy w.r.t. the Q-estimate. In this case, our loose bound of Corollary 1 scales similarly as the one obtained by [LWC+20b].

We conclude this section by noting that the advantage of the above upper bound $U_o(\mathcal{M})$ is that the corresponding allocation (that solves (8)) can be easily computed, and hence tracked. Indeed, the optimization problem in (8) has convex objective and constraints. Therefore, we can use the projected gradient-descent algorithm to compute $\boldsymbol{\omega}^\star = \arg\max_{\omega\in\Omega(\mathcal{M})} U(\mathcal{M}, \omega)^{-1}$ which will be used in our algorithm as a proxy for $\arg\max_{\omega\in\Omega(\mathcal{M})} T(\mathcal{M}, \omega)^{-1}$.

4 Algorithm

We propose MDP-NaS (MDP Navigate-and-Stop), a model-based algorithm that is inspired by the lower bound. Define the oracle allocation vector[4] $\boldsymbol{\omega}^\star(\mathcal{M}) \triangleq \arg\max_{\omega\in\Omega(\mathcal{M})} U(\mathcal{M}, \omega)$. The lower bound suggests that to identify the best policy in a sample-efficient manner, the algorithm must collect samples from state-action pair (s, a) proportionally to $\boldsymbol{\omega}^\star_{sa}(\mathcal{M})$. We propose two sampling rules which ensure that the former statement holds in the long term (see Section 4.1 for a rigorous formulation). Our sampling rules are combined with a Generalized Likelihood Ratio (GLR) test (or rather a proxy of the GLR test, see Section 4.2 for details), that stops as soon as we are confident that $\widehat{\pi}^\star_t = \pi^\star$ with probability at least $1 - \delta$.

MDP-NaS starts by drawing a random MDP with a unique optimal policy (for example, it can select Bernoulli rewards with means drawn from the uniform distribution on $[0, 1]$ and transitions from a Dirichlet distribution $\mathcal{D}(1, \ldots, 1)$) that will serve as an initial estimate $\widehat{\mathcal{M}}_0$ of $\mathcal{M}$). The algorithm maintains, after t time steps, an empirical estimate $\widehat{\mathcal{M}}_t$ of the true MDP. Based on this estimate, MDP-NaS computes the empirical oracle policy $\pi^o(\widehat{\mathcal{M}}_t)$ defined in (9), and selects the action to play next depending on the current state $a_t \sim \pi_t(.|s_t)$, where π_t is given by either of our sampling rules. After each observation, $\widehat{\mathcal{M}}_t$ is updated. Finally, the algorithm checks if the stopping condition in (12) is satisfied, in which case it stops and returns the empirical optimal policy $\widehat{\pi}^\star_\tau$.

4.1 Sampling rule

We introduce a few definitions to simplify the presentation. Any stationary policy π induces a Markov chain on $\mathcal{Z}$ whose transition kernel is defined by $P_\pi((s, a), (s', a')) \triangleq P(s'|s, a)\pi(a'|s')$. With

[4] Note that if we have access to an optimization oracle that, given a MDP $\mathcal{M}$, returns the optimal allocation solution to (4), then we can replace $\boldsymbol{\omega}^\star(\mathcal{M})$ by this optimal allocation. Our algorithm will then be asymptotically optimal up to a factor of 2.

some abuse of notation, we will use P_π to refer to both the Markov chain and its kernel. We denote by π_u the uniform random policy, i.e., $\pi_u(a|s) = 1/A$ for all pairs (s, a). Finally, we define the vector of visit-frequencies $\boldsymbol{N}(t)/t \triangleq \big(N_{sa}(t)/t\big)_{(s,a)\in\mathcal{Z}}$.

In contrast with pure exploration in Multi-Armed Bandits and MDPs with a generative model where any allocation vector in the simplex is achievable, here the agent can only choose a *sequence of behavior policies* and follow the resulting trajectory. Therefore, one might ask if the oracle allocation can be achieved by a following a simple policy. A natural candidate is the *oracle policy* defined by

$$\forall (s,a) \in \mathcal{Z}, \quad \pi^o(a|s) \triangleq \frac{\omega^\star_{sa}}{\omega^\star_s} = \frac{\omega^\star_{sa}}{\sum_{b\in\mathcal{A}} \omega^\star_{sb}}. \tag{9}$$

It is immediate to check that $\boldsymbol{\omega}^\star$ is the stationary distribution of P_{π^o}. Denote by $\pi^o(\mathcal{M})$ the above oracle policy defined through $\omega^\star(\mathcal{M})$. Policy $\pi^o(\mathcal{M})$ is the target that we would like to play, but since the rewards and dynamics of $\mathcal{M}$ are unknown, the actions must be chosen so as to estimate consistently the oracle policy while at the same time ensuring that $\boldsymbol{N}(t)/t$ converges to $\boldsymbol{\omega}^\star(\mathcal{M})$. The following two sampling rules satisfy these requirements.

D-Navigation rule: At time step t, the learner plays the policy $\pi_t = \varepsilon_t \pi_u + (1-\varepsilon_t)\pi^o(\widehat{\mathcal{M}}_t)$, where ε_t is a decreasing exploration rate to be tuned later.

C-Navigation rule: At time step t, the learner plays $\pi_t = \varepsilon_t \pi_u + (1-\varepsilon_t)\sum_{j=1}^t \pi^o(\widehat{\mathcal{M}}_j)/t$. Here the agent navigates using a Cesaro-mean of oracle policies instead of the current estimate of the oracle policy, which makes the navigation more stable.

4.1.1 Forced Exploration

The mixture with the uniform policy[5] helps the agent to explore all state-action pairs often enough in the initial phase. This is particularly necessary in the case where the empirical weight $\omega^\star_{sa}(\widehat{\mathcal{M}}_t)$ of some state-action pair (s, a) is under-estimated, which may cause that this pair is left aside and hence the estimation never corrected. The next result in this section gives a tuning of the rate ε_t that ensures a sufficient forced exploration:

Lemma 6. *Let m be the maximum length of the shortest paths (in terms of number of transitions) between pairs of states in $\mathcal{M}$: $m \triangleq \max_{(s,s')\in\mathcal{S}^2} \min\{n \geq 1 : \exists \pi : \mathcal{S} \to \mathcal{A}, P^n_\pi(s, s') > 0\}$. Then C-Navigation or D-Navigation with any decreasing sequence $(\varepsilon_t)_{t\geq 1}$ such that $\forall t \geq 1,\ \varepsilon_t \geq t^{-\frac{1}{m+1}}$ satisfies: $\mathbb{P}_{\mathcal{M},\mathbb{A}}\big(\forall (s,a) \in \mathcal{Z},\ \lim_{t\to\infty} N_{sa}(t) = \infty\big) = 1$.*

Remark 1. *When the parameter m is unknown to the learner, one can replace it by its worst case value $m_{\max} = S - 1$. However, when prior knowledge is available, using a faster-decreasing sequence $\varepsilon_t = t^{-\frac{1}{m+1}}$ instead of $t^{-\frac{1}{S}}$ can be useful to accelerate convergence, especially when the states of $\mathcal{M}$ are densely connected ($m \ll S - 1$).*

Minimal exploration rate: Communicating MDPs. The forced exploration rate $t^{-\frac{1}{S}}$ vanishes pretty slowly. One may wonder if this rate is necessary to guarantee sufficient exploration in communicating MDPs: the answer is yes in the worst case, as the following example shows. Consider a variant of the classical RiverSwim MDP with state (resp. action) space $\mathcal{S} = [|1, S|]$, (resp. $\mathcal{A} = \{\text{LEFT1}, \text{LEFT2}, \text{RIGHT}\}$). LEFT1 and LEFT2 are equivalent actions inducing the same rewards and transitions. After playing RIGHT the agent makes a transition of one step to the right, while playing LEFT1 (or LEFT2) moves the agent all the way back to state 1. The rewards are null everywhere but in states $\{1, S\}$ where $R(1, \text{LEFT1}), R(1, \text{LEFT2}))$ are equal to 0.01 almost surely and $R(S, \text{RIGHT})$ is Bernoulli with mean 0.02.

When γ is close to 1, the optimal policy consists in always playing RIGHT so as to reach state S and stay there indefinitely. Now suppose that the agent starts at $s = 1$ and due to the small probability of observing the large reward in state S, she underestimates the value of this state in the first rounds and focuses on distinguishing the best action to play in state 1 among $\{\text{LEFT1}, \text{LEFT2}\}$. Under this scenario she ends up playing the sequence of policies $\pi_t = 3\varepsilon_t \pi_u + (1 - 3\varepsilon_t)(\text{LEFT1} + \text{LEFT1})/2$. This induces the non-homogeneous Markov Chain depicted in Figure 1. For the exploration rate

[5] Our results still hold if π_u is replaced by any other policy π which has an ergodic kernel. In practice, this can be helpful especially when we have prior knowledge of a fast-mixing policy π.

$\varepsilon_t = t^{-\alpha}$, we show that if the agent uses any $\alpha > \frac{1}{S-1}$, with non-zero probability she will visit state S only a finite number of times. Therefore she will fail to identify the optimal policy for small values of the confidence level δ. Indeed if the agent visits state S at time k, then the last $S-1$ transitions before k

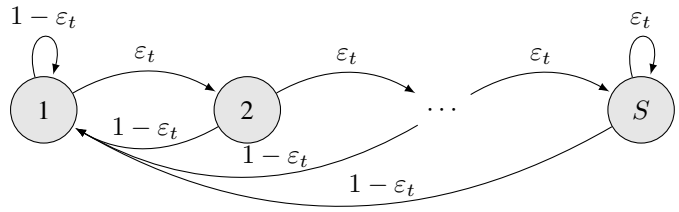

Figure 1: Non-homogeneous Markov Chain. An exploration rate of at least $t^{-\frac{1}{S-1}}$ is needed.

must have been to the right, ie $\mathbb{P}(s_k = S) \leq \prod_{j=k-S+1}^{k-1} \varepsilon_j \leq (\varepsilon_{k-S+1})^{S-1}$. Therefore $\mathbb{E}[N_S(t)] \leq \sum_{k=S}^{t} (k-S+1)^{-\alpha(S-1)}$. In particular this implies that for $\alpha > \frac{1}{S-1}$, $\limsup_{t\to\infty} \mathbb{E}[N_S(t)] = M < \infty$. Therefore using the reverse Fatou lemma and Markov's inequality:

$$\begin{aligned}\mathbb{P}(\forall t \geq 1, N_S(t) \leq 2M) &= \mathbb{E}\big[\limsup_{t\to\infty} \prod_{k=1}^{t} \mathbb{1}\{N_S(k) \leq 2M\}\big] \\ &\geq \limsup_{t\to\infty} \mathbb{E}\big[\prod_{k=1}^{t} \mathbb{1}\{N_S(k) \leq 2M\}\big] = \limsup_{t\to\infty} \mathbb{E}\big[\mathbb{1}\{N_S(t) \leq 2M\}\big] \geq \frac{1}{2}.\end{aligned}$$

Minimal exploration rate: Ergodic MDPs. For such MDPs, we can select $\varepsilon_t = 1/t^{\zeta}$ where $\zeta < 1$ without compromising the conclusion of Lemma 6. This is a consequence of Proposition 2 [BK97], stating that there exist $c_1, c_2, C > 0$ such that for all s and t large enough, $\mathbb{P}_{\mathcal{M},\mathbb{A}}[N_s(t) > c_1 t] \geq 1 - Ce^{-c_2 t}$. A union bound yields: $\mathbb{P}_{\mathcal{M},\mathbb{A}}[\forall s, N_s(t) > c_1 t] \geq 1 - CSe^{-c_2 t}$. To extend this result to the numbers of visits at the various (state, action) pairs, we can derive a lower bound on $N_{sa}(t)$ given that $N_s(t) > c_1 t$ by observing that a worst scenario (by monotonicity of ε_s) occurs when s is visited only in the $c_1 t$ rounds before t. We get $\mathbb{E}[N_{sa}(t) | N_s(t) > c_1 t] \geq c_3 t^{(1-\zeta)}$. Remarking that $N_{sa}(t+1) - N_s(t)\varepsilon_t$ is a sub-martingale with bounded increments, standard concentration arguments then imply that $\mathbb{P}_{\mathcal{M},\mathbb{A}}[\forall s, a, N_{sa}(t) > \frac{c_3}{2} t^{(1-\zeta)}] \geq \phi(t)$, where $\phi(t) \to 1$. Next, define the random variable $Z_t = \prod_{s,a} \mathbb{1}\{N_{sa}(t) > \frac{c_3}{2} t^{(1-\zeta)}\}$. Applying the reverse Fatou lemma, we get $1 = \limsup_t \mathbb{E}[Z_t] \leq \mathbb{E}[\limsup_t Z_t]$. From there, we directly deduce (by monotonicity of $t \mapsto N_{sa}(t)$) that a.s. $\lim_{t\to\infty} N_{sa}(t) = \infty$.

4.2 Stopping rule

To implement a GLR test, we define $\ell_{\mathcal{M}'}(t)$, the likelihood of the observations under some MDP $\mathcal{M}' \ll \mathcal{M}$: $\ell_{\mathcal{M}'}(t) = \prod_{k=0}^{t-1} p_{\mathcal{M}'}(s_{k+1}|s_k, a_k) q_{\mathcal{M}'}(R_k|s_k, a_k)$, where at step k the algorithm is in state s_k, plays action a_k and observes the reward R_k and s_{k+1} (the next state). Performing a GLR test in step t consists in computing the optimal policy $\widehat{\pi}_t^\star$ for the estimated MDP $\widehat{\mathcal{M}}_t$ and in comparing the likelihood of observations under the most likely model where $\widehat{\pi}_t^\star$ is optimal to the likelihood under the most likely model where $\widehat{\pi}_t^\star$ is sub-optimal. Following the stylized form of the GLR for multi-armed bandits in [DKM19] we write:

$$G_{\widehat{\pi}_t^\star}(t) = \log \frac{\sup_{\mathcal{M}':\widehat{\pi}_t^\star \in \Pi^\star(\mathcal{M}')} \ell_{\mathcal{M}'}(t)}{\sup_{\mathcal{M}':\widehat{\pi}_t^\star \notin \Pi^\star(\mathcal{M}')} \ell_{\mathcal{M}'}(t)} = \log \frac{\ell_{\widehat{\mathcal{M}}_t}(t)}{\sup_{\mathcal{M}' \in \mathrm{Alt}(\widehat{\mathcal{M}}_t)} \ell_{\mathcal{M}'}(t)} = \inf_{\mathcal{M}' \in \mathrm{Alt}(\widehat{\mathcal{M}}_t)} \log \frac{\ell_{\widehat{\mathcal{M}}_t}(t)}{\ell_{\mathcal{M}'}(t)}$$

$$= \inf_{\mathcal{M}' \in \mathrm{Alt}(\widehat{\mathcal{M}}_t)} \sum_{(s,a)\in\mathcal{Z}} N_{sa}(t) \big[\mathrm{KL}(\widehat{q}_{s,a}(t),\, q_{\mathcal{M}'}(s,a)) + \mathrm{KL}(\widehat{p}_{s,a}(t),\, p_{\mathcal{M}'}(s,a)) \big] \quad (10)$$

$$= t\, T\Big(\widehat{\mathcal{M}}_t, \boldsymbol{N}(t)/t\Big)^{-1}. \quad (11)$$

The hypothesis ($\widehat{\pi}_t^\star \neq \pi^\star$) is then rejected as soon as the ratio of likelihoods becomes greater than the threshold $\beta(t,\delta)$, properly tuned to ensure that the algorithm is δ-PC. Note that the likelihood ratio $G_{\widehat{\pi}_t^\star}(t)$ itself can be difficult to compute for MDPs, since it is equivalent to solving (5), we lower bound it using (11) and Lemma 5. This leads to the following proposition:

Proposition 7. *Define the random thresholds for the transitions and rewards respectively:*

$$\beta_p(t,\delta) \triangleq \log(1/\delta) + (S-1)\sum_{s,a} \log\Big[e\big(1 + N_{sa}(t)/(S-1)\big)\Big]$$

$$\beta_r(t,\delta) \triangleq SA\, \varphi\big(\log(1/\delta)/SA\big) + 3\sum_{s,a} \log\big[1 + \log(N_{sa}(t))\big]$$

where $\varphi(x) \underset{\infty}{\sim} x + \log(x)$ is defined in the appendix. Then the stopping rule:

$$\tau_\delta \triangleq \inf\Big\{ t \geq 1 :\ t\, U\big(\widehat{\mathcal{M}}_t, \boldsymbol{N}(t)/t\big)^{-1} \geq \beta_r(t,\delta/2) + \beta_p(t,\delta/2) \Big\} \quad (12)$$

is δ-PC, i.e., $\mathbb{P}(\tau_\delta < \infty,\ \widehat{\pi}_{\tau_\delta}^\star \neq \pi^\star) \leq \delta$.

5 Main Results and Sample Complexity Analysis

First we state our main results, which take the form of asymptotic bounds on the sample complexity of MDP-NaS, under a slightly more conservative exploration rate than Lemma 6. Then we present the most important ingredients in their proof. The complete proof is provided in the appendix.

Theorem 8. *Using the C-Navigation sampling rule with $\varepsilon_t = t^{-\frac{1}{2(m+1)}}$ and the stopping rule (12): (i) MDP-NaS stops almost surely and its stopping time satisfies $\mathbb{P}\big(\limsup_{\delta\to 0} \frac{\tau_\delta}{\log(1/\delta)} \leq 2U_o(\mathcal{M})\big) = 1$, where $U_o(\mathcal{M})$ was defined in (8); (ii) the stopping time of MDP-NaS has a finite expectation for all $\delta \in (0,1)$, and $\limsup_{\delta\to 0} \frac{\mathbb{E}[\tau_\delta]}{\log(1/\delta)} \leq 2U_o(\mathcal{M})$.*

5.1 Proof sketch

Concentration of empirical MDPs: The starting point of our proof is a concentration event of the empirical estimates $\widehat{\mathcal{M}}_t$ around $\mathcal{M}$. For $\xi > 0$ and $T \geq 1$, we define $\mathcal{C}_T^1(\xi) = \bigcap_{t=T^{1/4}}^{T} \left(\left\|\widehat{\mathcal{M}}_t - \mathcal{M}\right\| \leq \xi, \left\|\pi^o(\widehat{\mathcal{M}}_t) - \pi^o(\mathcal{M})\right\|_\infty \leq \xi \right)$, where $\|.\|$ is a semi-norm on MDPs. Then we prove in Lemma 16 that $\mathcal{C}_T^1(\xi)$ holds with high probability in the sense that for all $T \geq 1$:

$$\mathbb{P}\left(\mathcal{C}_T^1(\xi)\right) \geq 1 - \mathcal{O}\big(1/T^2\big). \quad (13)$$

For this purpose we derive, for all pairs (s,a), a lower bound on $N_{sa}(t)$ stating that[6]: $\mathbb{P}\big(\forall (s,a) \in \mathcal{Z},\ \forall t \geq 1,\ N_{sa}(t) \geq (t/\lambda_\alpha)^{1/4} - 1\big) \geq 1 - \alpha$ where $\lambda_\alpha \propto \log^2(1 + SA/\alpha)$ is a

[6]Using the method in our proof one can derive a lower bound of the type t^ζ for any $\zeta < 1/2$ and any exploration rate $\varepsilon_t = t^{-\theta}$ with $\theta < 1/(m+1)$. We chose $\theta = \frac{1}{2(m+1)}$ and $\zeta = 1/4$ because they enable us to have an explicit formula for λ_α. See Corollary 2 in the appendix.

parameter that depends on the mixing properties of $\mathcal{M}$ under the uniform policy. These lower bounds on $N_{sa}(t)$ w.h.p. contrast with their deterministic equivalent obtained for C-tracking in [GK16].

Concentration of visit-frequencies: Before we proceed, we make the following assumption.

Assumption 2. *P_{π_u} is aperiodic*[7].

Under Assumptions 1 and 2 the kernel P_{π_u}, and consequently also P_{π^o}, becomes ergodic. Hence the iterates of P_{π^o} converge to $\boldsymbol{\omega}^\star(\mathcal{M})$ at a geometric speed. Also note that the Markov chains induced by playing either of our sampling rules are non-homogeneous, with a sequence of kernels $(P_{\pi_t})_{t\geq 1}$ that is history-dependent. To tackle this difficulty, we adapt a powerful ergodic theorem (see Proposition 13 in the appendix) from [FMP11] originally derived for adaptive Markov Chain Monte-Carlo (MCMC) algorithms to get the following result.

Theorem 9. *Using the C-Navigation or D-Navigation we have: $\lim_{t\to\infty} \boldsymbol{N}(t)/t = \omega^\star(\mathcal{M})$ almost surely.*

Lemma 6 and Theorem 9 combined prove Theorem 8 *(i)* in a straightforward fashion. The proof of Theorem 8 *(ii)* is more involved. Again, we adapt the proof method from [FMP11] to derive a finite-time version of Proposition 13 which results into the following proposition.

Proposition 10. *Under C-Navigation, for all $\xi > 0$, there exists a time T_ξ such that for all $T \geq T_\xi$, all $t \geq T^{3/4}$ and all functions $f : \mathcal{Z} \to \mathbb{R}^+$, we have:*

$$\mathbb{P}\left(\left|\frac{\sum_{k=1}^t f(s_k,a_k)}{t} - \mathbb{E}_{(s,a)\sim\omega^\star}[f(s,a)]\right| \geq K_\xi \left\|f\right\|_\infty \xi \,\middle|\, \mathcal{C}_T^1(\xi)\right) \leq 2\exp\big(-t\xi^2\big).$$

where $\xi \mapsto K_\xi$ is a mapping with values in $(1,\infty)$ such that $\limsup_{\xi\to 0} K_\xi < \infty$.

Now define $\mathcal{C}_T^2(\xi) = \bigcap_{t=T^{3/4}}^T \big(\big|\boldsymbol{N}(t)/t - \omega^\star(\mathcal{M})\big| \leq K_\xi\ \xi\big)$. Then Proposition 10 and Eq. (13) combined imply that for T large enough, the event $\mathcal{C}_T^1(\xi) \cap \mathcal{C}_T^2(\xi)$ holds w.h.p. so that the expected stopping time is finite on the complementary event: $\mathbb{E}[\tau_\delta \mathbb{1}\{\overline{\mathcal{C}_T^1(\xi)} \cup \overline{\mathcal{C}_T^2(\xi)}\}] < \infty$. Now given the asymptotic shape of the thresholds: $\beta_r(\delta,t) + \beta_p(\delta,t) \underset{\delta\to 0}{\sim} 2\log(1/\delta)$, we may informally write:

$$\mathbb{E}\big[\tau_\delta \mathbb{1}\{\mathcal{C}_T^1(\xi)\cap\mathcal{C}_T^2(\xi)\}\big] \underset{\delta\to 0}{\preceq} 2\log(1/\delta) \sup_{(\mathcal{M}',\omega')\in B_\xi} U\big(\mathcal{M}',\omega'\big),$$

where $B_\xi = \{(\mathcal{M}',\omega') : \|\mathcal{M}' - \mathcal{M}\| \leq \xi, \|\omega' - \omega^\star(\mathcal{M})\|_\infty \leq K_\xi\ \xi\}$. Taking the limits when δ and ξ goes to zero respectively concludes the proof.

6 Conclusion

To the best of our knowledge, this paper is the first to propose an algorithm with *instance-dependent sample complexity* for Best Policy identification (BPI) in the *full online* setting. Our results are encouraging as they show: 1) How the navigation constraints of online RL impact the difficulty of learning a good policy, compared to the more relaxed sampling schemes of Multi-Armed Bandits and MDPs with a generative model. 2) That, provided access to an optimization oracle that solves the (resp. some convex relaxation of the) information-theoretical lower bound, asymptotic optimal (resp. near-optimal) sample complexity is still possible through adaptive control of the trajectory. This opens up exciting new research questions. First, it is intriguing to understand how the mixing times -and not just the stationary distributions- of Markov chains induced by policies impact the sample complexity of BPI in the moderate confidence regime. A second direction would be to extend our contributions to the problem of finding an ε-optimal policy, which is of more practical interest than identifying the best policy.

[7] This assumption is mild as it is enough to have only one state $\tilde{s}$ and one action $\tilde{a}$ such that $P_{\mathcal{M}}(\tilde{s}|\tilde{s},\tilde{a}) > 0$ for it to be satisfied. Furthermore, assumptions 1 and 2 combined are still less restrictive than the usual *ergodicity* assumption, which requires that the Markov chains of *all* policies be ergodic.

References

[AHKS20] Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13399–13412. Curran Associates, Inc., 2020.

[AJO09] Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2009.

[AKY20] Alekh Agarwal, Sham Kakade, and Lin F. Yang. Model-based reinforcement learning with a generative model is minimax optimal. volume 125 of *Proceedings of Machine Learning Research*, pages 67–83. PMLR, 09–12 Jul 2020.

[AMK13] Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J Kappen. Minimax pac bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning*, 91(3):325–349, 2013.

[BK97] Apostolos N. Burnetas and Michael N. Katehakis. Optimal adaptive policies for markov decision processes. *Mathematics of Operations Research*, 22(1):222–255, 1997.

[Din21] Yuhao Ding. Global convergence of stochastic policy gradient methods with momentum and entropy. 2021.

[DKM19] Rémy Degenne, Wouter M. Koolen, and Pierre Ménard. Non-asymptotic pure exploration by solving games. In *NeurIPS*, 2019.

[DMKV21] O. D. Domingues, Pierre M'enard, E. Kaufmann, and Michal Valko. Episodic reinforcement learning in finite mdps: Minimax lower bounds revisited. In *ALT*, 2021.

[DTC13] Thomas Dietterich, Majid Taleghan, and Mark Crowley. Pac optimal planning for invasive species management : Improved exploration for reinforcement learning from simulator-defined mdps. 01 2013.

[EDMM06] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 06 2006.

[FCG10] Sarah Filippi, Olivier Cappé, and Aurélien Garivier. Optimism in reinforcement learning and kullback-leibler divergence. In *Allerton Conference on Communication, Control, and Computing*, pages 115–122, Sep. 2010.

[Fie94] C. Fiechter. Efficient reinforcement learning. In *COLT '94*, 1994.

[FMP11] G. Fort, É. Moulines, and P. Priouret. Convergence of adaptive and interacting markov chain monte carlo algorithms. *Annals of Statistics*, 39:3262–3289, 2011.

[FYAY21] Fei Feng, W. Yin, Alekh Agarwal, and Lin F. Yang. Provably correct optimization and exploration with non-linear policies. *ArXiv*, abs/2103.11559, 2021.

[GK16] Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 998–1027, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

[GK19] A. Garivier and Emilie Kaufmann. Non-asymptotic sequential tests for overlapping hypotheses and application to near optimal arm identification in bandit models. *arXiv preprint, Statistics Theory*, arXiv:1905.03495, 2019.

[Kak03] Sham Machandranath Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, University of London, England, 2003.

[KCG16] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.

[KMDD+21] Emilie Kaufmann, Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Edouard Leurent, and Michal Valko. Adaptive reward-free exploration. In Vitaly Feldman, Katrina Ligett, and Sivan Sabato, editors, *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pages 865–891. PMLR, 16–19 Mar 2021.

[KS98] Michael Kearns and Satinder Singh. Finite-sample convergence rates for q-learning and indirect algorithms. In *Proceedings of the 11th International Conference on Neural Information Processing Systems*, NIPS'98, page 996–1002, Cambridge, MA, USA, 1998. MIT Press.

[KS99] Michael Kearns and Satinder Singh. Finite-sample convergence rates for q-learning and indirect algorithms. *Advances in Neural Information Processing*, 11, 04 1999.

[LCC+21] Gen Li, Changxiao Cai, Yuxin Chen, Yuantao Gu, Yuting Wei, and Yuejie Chi. Is q-learning minimax optimal? a tight sample complexity analysis, 2021.

[LR85] T.L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–2, 1985.

[LS20] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[LWC+20a] Gen Li, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen. Breaking the sample size barrier in model-based reinforcement learning with a generative model. *arXiv preprint*, arXiv:2005.12900, 2020.

[LWC+20b] Gen Li, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen. Sample complexity of asynchronous q-learning: Sharper analysis and variance reduction, 2020.

[MDJ+20] Pierre M'enard, O. D. Domingues, Anders Jonsson, E. Kaufmann, Edouard Leurent, and Michal Valko. Fast active learning for pure exploration in reinforcement learning. *ArXiv*, abs/2007.13442, 2020.

[MP20] Aymen Al Marjani and Alexandre Proutiere. Adaptive sampling for best policy identification in markov decision processes. *arXiv preprint*, arXiv:2009.13405, 2020.

[Rus16] D. Russo. Simple bayesian algorithms for best arm identification. In *Proceedings of the 29th Conference On Learning Theory*, 2016.

[SB18] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[SWW+18] Aaron Sidford, Mengdi Wang, Xian Wu, Lin Yang, and Yinyu Ye. Near-optimal time and sample complexities for solving markov decision processes with a generative model. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5186–5196. Curran Associates, Inc., 2018.

[Sze10] Csaba Szepesvari. *Algorithms for Reinforcement Learning*. Morgan and Claypool Publishers, 2010.

[TL19] Jean Tarbouriech and Alessandro Lazaric. Active exploration in markov decision processes. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 974–982. PMLR, 16–18 Apr 2019.

[Wan17] Mengdi Wang. Randomized Linear Programming Solves the Discounted Markov Decision Problem In Nearly-Linear (Sometimes Sublinear) Running Time. *arXiv e-prints*, page arXiv:1704.01869, April 2017.

[ZCA21] Andrea Zanette, Ching-An Cheng, and Alekh Agarwal. Cautiously optimistic policy optimization and exploration with linear function approximation. *ArXiv*, abs/2103.12923, 2021.

[ZKB19] Andrea Zanette, Mykel J Kochenderfer, and Emma Brunskill. Almost horizon-free structure-aware best policy identification with a generative model. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5625–5634. Curran Associates, Inc., 2019.

[ZKOB20] Junzi Zhang, J. Kim, Brendan O'Donoghue, and S. Boyd. Sample efficient reinforcement learning with reinforce. *ArXiv*, abs/2010.11364, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] In the conclusion.
   (c) Did you discuss any potential negative societal impacts of your work? **[TODO]**
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] In the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] see Readme file in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] see supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [No] The code and data are proprietary.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [No] No crowd-sourcing or research with human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]