# Finite-Memory Near-Optimal Learning for Markov Decision Processes with Long-Run Average Reward

**Jan Kretínský**
TU Munich
Munich, Germany

**Fabian Michel**
TU Munich
Munich, Germany

**Lukas Michel**
TU Munich
Munich, Germany

**Guillermo A. Pérez**
University of Antwerp
Antwerp, Belgium

## Abstract

We consider learning policies online in Markov decision processes with the long-run average reward (a.k.a. mean payoff). To ensure implementability of the policies, we focus on policies with finite memory. Firstly, we show that near optimality can be achieved almost surely, using an unintuitive gadget we call forgetfulness. Secondly, we extend the approach to a setting with partial knowledge of the system topology, introducing two optimality measures and providing near-optimal algorithms also for these cases.

## 1 INTRODUCTION

**Markov decision processes** (MDP, e.g. Puterman (2005)) is the standard formalism for modelling and analysis of sequential decision making in the presence of probabilistic uncertainty. Their applications range from robot motion planning (Russell and Norvig, 2010), modelling and playing Go (Gibney, 2016), to scheduling (Geeraerts et al., 2018). Learning (near-)optimal policies (a.k.a. strategies) for MDPs has been investigated in the context of various optimization objectives, such as discounted reward (Watkins and Dayan, 1992), total reward (Dann and Brunskill, 2015), and long-run average reward (Brafman and Tennenholtz, 2002).

**Long-run average reward** (a.k.a. mean payoff) is an infinite-horizon objective that, in contrast to discounted reward, requires considering infinite runs even for approximating the optimal value. This makes learning less efficient, the whole problem more difficult and also less studied in the context of AI or robotics. However, average reward captures in a much more adequate way the performance over an unknown or variable horizon than the discounted one (see, e.g. Schwartz (1993)).

*Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, PMLR volume 124, 2020.

**Learning** (near-)optimal strategies has two main advantages over model-based analysis of MDPs. Firstly, it is applicable also to (partially) *unknown* systems. Secondly, it can be more *practical* as it can scale better than an analysis of the whole state space. These two advantages imply the following two issues that should thus be reflected by the learning framework and techniques.

(1) The system may be entirely unknown, in which case we cannot even ensure any safety properties of the behaviour. Alternatively, it may be partially known: typically the quantities (exact transition probabilities) are unknown, but the qualitative topology may be known (which actions may lead to which states). In this paper, we consider a more flexible option, where the *topology is partially known*, i.e. presence or absence of transitions between states may or may not be determined. For example, consider an action, which (i) can fail in several known ways (with unknown probabilities), (ii) certainly cannot affect unrelated components of the system, but (iii) has unknown impact on other aspects. We introduce *modal MDPs* to capture such knowledge.

(2) The strategies produced may require infinite memory in order to be optimal with high probability. Such strategies are not *practical to implement*. We thus restrict the scope of the problem to *finite-memory strategies*. In contrast, R-MAX (Brafman and Tennenholtz, 2002) and UCRL2 (Jaksch et al., 2010) continue updating their statistics forever: For their optimality guarantees to hold they have to be run to infinity. Since, at every step they update internal variables based on new data, their policies have access to an unbounded amount of storage or variables with unbounded precision, i.e. so-called *infinite memory*.

**Our contributions** can be summarized as follows:

- We argue that PAC learning is more appropriate in this context than regret minimization.
- We provide the first optimal finite-memory strategy for PAC online learning in MDPs with long-run average reward, via a gadget we call forgetfulness.

- We extend the approach to modal MDPs. In contrast to many other works, we do not assume any restriction on the structure of the system such as strong connectedness, unichains etc.

**Related work.** Our work resembles earlier model-based learning algorithms for long-run average reinforcement learning like (Jaksch et al., 2010; Auer and Ortner, 2006). Nevertheless, we focus on PAC-like bounds since we work towards finite-memory strategies. That means that bounding regret (like in previous works), which would imply the PAC bounds, is not an option (see Corollary 1). The closest to our work is (Křetínský et al., 2018), which considers PAC online learning for MDPs with MP. However, full knowledge of the topology is assumed there. Besides, we identify an incorrect statement (Křetínský et al., 2018, Proposition 9) claiming impossibility of almost-sure near-optimality for finite-memory strategies; we correct the claim, showing the opposite by introducing "forgetful" strategies.

Related to learning for MDPs is statistical model checking (SMC) for MDPs, which constructs a strategy using possibly more (re-started) runs of the system. SMC of unbounded-horizon properties of MDPs was first considered in (Lassaigne and Peyronnet, 2012; Henriques et al., 2012). (Hahn et al., 2019) gives a convergent model-free algorithm (with no bounds on the current error) and identifies errors in some previous approaches.

Several approaches provide SMC for MDPs and unbounded properties with *PAC guarantees*. Firstly, the algorithm of (Fu and Topcu, 2014) requires (1) the mixing time $T$ of the MDP, (2) the ability to restart simulations also in non-initial states, (3) visiting *all* states sufficiently many times, and thus (4) the size of the state space $|S|$. Secondly, (Brázdil et al., 2014), based on delayed Q-learning (Strehl et al., 2006), lifts the assumptions (2) and (3) and instead of (1) requires only (a bound on) the minimum transition probability $\mathbf{p}_{\min}$. Thirdly, (Ashok et al., 2019) additionally lifts the assumption (4), keeping only $\mathbf{p}_{\min}$, as in this paper.

In (Daca et al., 2016), it is argued that while unbounded properties cannot be analysed without any information on the system, knowledge of (a lower bound on) the minimum transition probability $\mathbf{p}_{\min}$ is a relatively light and realistic assumption in many scenarios, in particular compared to the knowledge of the whole topology. In this paper, we thus adopt this assumption.

# 2 PRELIMINARIES

For a countable set $S$, we denote by $\mathrm{Dist}(S)$ the set of all *(rational) probabilistic distributions* on $S$, that is the set of all functions $f : S \to [0,1] \cap \mathbb{Q}$ such that $\sum_{s \in S} f(s) = 1$. For a set $X$ and a function $g : X \to \mathrm{Dist}(S)$, we write $g(s|x)$ instead of $g(x)(s)$. The *support* of a distribution $f \in \mathrm{Dist}(S)$ is the set $\mathrm{supp}(f) \stackrel{\text{def}}{=} \{s \in S : f(s) > 0\}$; for a function $g : X \to \mathrm{Dist}(S)$, we set $\mathrm{supp}(g) \stackrel{\text{def}}{=} \{(x,s) \in X \times S : g(s|x) > 0\}$ and for a relation $h \subseteq X \times S$, $\mathrm{supp}(h,x) \stackrel{\text{def}}{=} \{s \in S : (x,s) \in h\}$.

## 2.1 MARKOV CHAINS

**Definition 1.** *A* Markov chain (MC) *is a tuple* $\mathcal{C} = (Q, P, R)$ *where* $Q$ *is a countable set of states,* $P$ *is a probabilistic transition function* $P : Q \to \mathrm{Dist}(Q)$*, and* $R : Q \times Q \to [0,1] \cap \mathbb{Q}$ *is a reward function.*

A *run* of an MC is an infinite sequence of states $q_0 q_1 \ldots$ such that $P(q_{i+1}|q_i) > 0$ for all $i \geq 0$. For an initial state $q_0$, $\mathrm{Runs}^{q_0}(\mathcal{C})$ denotes the set of all runs of $\mathcal{C}$ that start with the state $q_0$ and $\Pr^{q_0}_{\mathcal{C}}[\cdot]$ is the unique probability measure respecting $P$ by Carathéodory's extension theorem (Puterman, 2005). Further, for a measurable function $f : \mathrm{Runs}^{q_0}(\mathcal{C}) \to \mathbb{R}$, we write $\mathrm{E}^{q_0}_{\mathcal{C}}[f]$ for the *expected value* of the function $f$ under the probability measure $\Pr^{q_0}_{\mathcal{C}}[\cdot]$ (Puterman, 2005).

**Mean payoff** is the random variable $\mathbf{MP}$ assigning to each run $\rho = q_0 q_1 \ldots$ of $\mathcal{C}$ the value

$$\mathbf{MP}(\rho) \stackrel{\text{def}}{=} \liminf_{n \in \mathbb{N}_{>0}} \mathbf{Avg}_n(\rho),$$

$$\text{where} \qquad \mathbf{Avg}_n(\rho) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=0}^{n-1} R(q_i, q_{i+1}).$$

Both $\mathbf{Avg}_n$ (for all $n$) and $\mathbf{MP}$ are Borel definable, thus also measurable, and therefore the *expected mean payoff* $\mathrm{E}^{q_0}_{\mathcal{C}}[\mathbf{MP}]$ is well defined and finite.

## 2.2 MARKOV DECISION PROCESSES

**Definition 2.** *A* Markov decision process (MDP) *is a tuple* $\mathcal{M} = (Q, q_0, A, P, R)$ *where* $Q$ *is a finite set of states,* $q_0 \in Q$ *is the initial state,* $A$ *is a finite set of actions,* $P : Q \times A \nrightarrow \mathrm{Dist}(Q)$ *is a partial probabilistic transition function, and* $R : Q \times A \times Q \to [0,1] \cap \mathbb{Q}$ *is a reward function.*

Let $A(q) \subseteq A$ denote the subset of *available* actions from state $q$, that is, the set of all actions $a \in A$ such that $P(q,a)$ is defined. We make the assumption that MDPs contain no deadlocks (a.k.a. dead-ends), i.e. $A(q) \neq \varnothing$ for all $q \in Q$.

A *history* $h$ of an MDP is a finite state-action-reward sequence $q_0 a_0 r_0 \ldots a_{k-1} r_{k-1} q_k$ such that $a_i \in A(q_i)$, $P(q_{i+1}|q_i, a_i) > 0$, and $r_i = R(q_i, a_i, q_{i+1})$, for all $0 \leq i < k$. We write $\mathrm{last}(h)$ to denote $q_k$; $\mathrm{Hist}^q(\mathcal{M})$ for the set of all histories of $\mathcal{M}$ starting with $q$.

**Definition 3.** *A* strategy $\sigma$ *in an MDP* $\mathcal{M} = (Q, q_0, A, P, R)$ *is a function* $\sigma : \mathrm{Hist}^{q_0}(\mathcal{M}) \to \mathrm{Dist}(A)$ *such that* $\sigma(a|h) > 0$ *implies that* $a \in A(\mathrm{last}(h))$.

**Semantics of MDP.** A strategy $\sigma$ for an MDP $\mathcal{M}$ resolves all nondeterministic choices of actions and thus yields an MC $\mathcal{M}^\sigma = (\mathrm{Hist}^{q_0}(\mathcal{M}), P^\sigma, R^\sigma)$ where

$$P^\sigma(h'|h) = \begin{cases} \sigma(a|h) \cdot P(q'|\mathrm{last}(h), a) & \text{if } h' = harq' \\ 0 & \text{otherwise} \end{cases}$$

Similarly, $R^\sigma(h, h')$ is $r$ if $h' = harq'$ and 0 otherwise.

**Types of strategies.** A strategy $\sigma$ is called *memoryless* if for all histories $h, h'$ we have that $\mathrm{last}(h) = \mathrm{last}(h')$ implies $\sigma(h) = \sigma(h')$; it is *deterministic* if for all histories $h$ the distribution $\sigma(h)$ is Dirac.

Further, we are interested in strategies implementable using real-world hardware or software with finite resources, in particular with *finite memory*. These strategies are implementable by a stochastic Mealy machine:

**Definition 4** (Finite-memory strategies)**.** *A* stochastic Mealy machine *for an MDP* $\mathcal{M} = (Q, q_0, A, P, R)$ *is a tuple* $\mathcal{T} = (M, m_0, f_u, f_o)$ *where* $M$ *is a finite set of memory elements,* $m_0 \in M$ *is the initial memory element,* $f_u : M \times Q \times A \times \mathbb{Q} \to M$ *is an update function, and* $f_o : M \times Q \to \mathrm{Dist}(A)$ *is an output function. We denote by* $\sigma_\mathcal{T}$ *the strategy implemented by* $\mathcal{T}$ *as follows: for all histories* $h = q_0 a_0 r_0 \dots a_{k-1} r_{k-1} q_k$ *we have* $\sigma_\mathcal{T}(h) = f_o(m_k, q_k)$*, where* $m_k$ *is inductively defined as* $m_{i+1} = f_u(m_i, q_i, a_i, r_i)$ *for all* $i \geq 1$*. We call such strategies* finite-memory strategies*.*

**End components** Here we recall the fundamental notion for MDP analysis.

**Definition 5.** *An* end component *(EC) of an MDP* $\mathcal{M} = (Q, q_0, A, P, R)$ *is a pair* $(S, B)$ *with* $S \subseteq Q$*,* $B \subseteq A$ *s.t.*

- *for all* $s \in S$ *and all* $a \in B(s)$ *we have* $\mathrm{supp}(P(s, a)) \subseteq S$*, and*
- *the directed graph with nodes from* $S$ *and edges* $\{(s, s') \in S \times S : \exists a \in B(s), s' \in \mathrm{supp}(P(s, a))\}$ *is strongly connected.*

Intuitively, an EC is a subsystem where one can stay forever and see all of it infinitely often.

### 2.3 ASSUMPTIONS

In our online-learning setting, we do not assume the complete knowledge of the MDP $\mathcal{M} = (Q, q_0, A, P, R)$ but only the following.

**Executability.** Intuitively, we assume we can run the MDP. Formally, we know the initial state $q_0$; given any state $q$, we know the set $A(q)$ of its available actions; given a state $q$ and an available action $a$, we can sample the successor according to $P(\cdot \mid q, a)$, observe the sampled successor $q'$ and the respective reward $R(q, a, q')$. For modal MDPs defined later, this assumption allows us to avoid strange questions such as: "If the transition is not really there, can I still play the action labelling it?"

**Partial knowledge of topology.** While we do not necessarily assume anything about the topology of the MDP, we want to reflect the partial knowledge whenever we have any. In order to formalize this, we assume we know an *under-approximation* $\underline{T}$ and an *over-approximation* $\overline{T}$ of the transition relation, i.e., relations satisfying $\underline{T} \subseteq \mathrm{supp}(P) \subseteq \overline{T} \subseteq Q \times A \times Q$. In other words, the transitions of $\underline{T}$ *must* be present with nonzero probability, the transitions of $\overline{T} \setminus \underline{T}$ *may* be present (it is not known), and the transitions of $Q \times A \times Q \setminus \overline{T}$ certainly have probability 0. We also use terms *must transitions* and *may transitions* according to the tradition of modal transition systems (Larsen and Thomsen, 1988). W.l.o.g., for all states $q$, $A(q) = \{a \in A : \exists q' \in Q : (q, a, q') \in \overline{T}\}$.

**Minimum nonzero probability.** Here we introduce our main restrictive assumption, used in all our results necessarily due to our finite-memory restriction. We assume the knowledge of a *lower bound* for all nonzero transition probabilities, i.e. a $\mathbf{p}_{\min} \in \mathbb{Q}$ such that $0 < \mathbf{p}_{\min} \leq P(q'|q, a)$ for all $(q, a, q') \in \mathrm{supp}(P)$. See Fig. 4 in Section 6 for an example of a MDP which demonstrates the need for $\mathbf{p}_{\min}$.

### 2.4 PROBLEM STATEMENT

The fundamental problem is to obtain strategies that maximize the expected mean payoff, (nearly) achieving the value $\mathbf{Val}(\mathcal{M}) \stackrel{\text{def}}{=} \sup_\sigma \mathrm{E}^{q_0}_{\mathcal{M}^\sigma}[\mathbf{MP}]$. In our setting there are two particular aspects to be taken into account. Firstly, we are interested in *finite-memory* strategies. Fortunately, the supremum is realized by a maximum over the memoryless strategies (Puterman, 2005), hence this is not a real restriction and the strategies we produce are near-optimal among all strategies.

Secondly, we obtain the strategies by *online learning with the limited knowledge* of the MDP. In Section 3, we investigate the case of strongly connected systems (applies to unichains, too), where we can guarantee the optimum. In general, the MDP can have more *maximal ECs* (MECs). Since different MECs are not mutually reachable with probability 1, we have to resolve online (with the limited knowledge) in which MEC we remain, and thus it is impossible to guarantee the optimal value. Instead, we optimize the mean payoff *given* that we remain in a certain EC. We examine the notions of ECs in the online-learning context and the corresponding optimization problems in the subsequent sections.

## 3 FORGETFUL LEARNING

In this section, we tackle a simpler problem, where the graph topology of the MDP is known and strongly connected, as stated by the following two assumptions:

1. We assume $\underline{T} = \overline{T}$, i.e., we know which transitions have positive probability.
2. The whole MDP $\mathcal{M}$ constitutes an *end component*.

Recall that for every MDP, there always exists an optimal strategy that is memoryless (Gimbert, 2007), i.e., a memoryless strategy $\tau$ such that $\mathrm{E}^{q_0}_{\mathcal{M}^\tau}[\mathbf{MP}] = \mathbf{Val}(\mathcal{M})$. We present a family of online-learnt finite-memory strategies $\sigma_\varepsilon$ that ensure, given any $\varepsilon \in (0,1)$, a mean payoff that is $\varepsilon$-close to the optimum $\mathbf{Val}(\mathcal{M})$.

The strategy $\sigma_\varepsilon$ plays in *episodes* sub-divided into repeated *exploration* and *exploitation phases* as follows.

- **Explore:** First, $\sigma_\varepsilon$ chooses actions uniformly at random during $L$ (stands for $L$earning) steps to collect statistics allowing it to compute empirical approximations $\widehat{P}$ and $\widehat{R}$ of $P$ and $R$.
- **Exploit:** Second, $\sigma_\varepsilon$ follows a (memoryless) expectation-optimal strategy $\tau$ for $\widehat{\mathcal{M}} = (Q, q_0, A, \widehat{P}, \widehat{R})$ during $O$ (stands for $O$ptimization) steps.
- Then the strategy $\sigma_\varepsilon$ "forgets the learnt model" and restarts from the exploration phase.

Since $\sigma_\varepsilon$ keeps information for a finite number of steps, the strategy can be implemented with finite memory.

Intuitively, it is clear that one round of long enough exploration and exploitation forever establishes a precise enough model with high enough probability, yielding PAC guarantees. In order to achieve almost-sure guarantees, the forgetting and restarting is fundamental. Without it, too high imprecision still has positive probability. The following result tells us that the repetition limits the probability of too high imprecision to zero. Note that Křetínský et al. (2018) falsely claims this is impossible.

**Theorem 1.** *For all $\varepsilon \in (0,1)$, one can compute $L, O \in \mathbb{N}$ such that for the resulting finite-state strategy $\sigma_\varepsilon$ we have $\Pr^{q_0}_{\mathcal{M}^{\sigma_\varepsilon}}[\rho : \mathbf{MP}(\rho) \geq \mathbf{Val}(\mathcal{M}) - \varepsilon] = 1$.*

*Proof idea.*

- By Hoeffding's inequality, we can easily compute a number $L$ such that $L$ steps of the exploration phase yield a good approximation of the dynamics of the MDP with high probability.
- We show that $O$ steps of the exploitation phase are sufficient to compensate for the learning phase so that the each episode has near-optimal expected mean payoff w.h.p. The computation of $O$ is considerably more involved. Since we do not have access to the actual transition probabilities, it relies on robustness (a.k.a. simulation (Kearns and Singh, 2002)) lemmas for mean payoff. Furthermore, the finite-memory restriction complicates using probabilistic guarantees on the convergence to the stationary distribution (such as Tracol, 2009). We circumvent this problem by using *exact-mixing stopping-time algorithms* for unknown Markov chains (See, e.g., Lovász and Winkler, 1995; Propp and Wilson, 1998) and the fact that their expectation can be bounded as function of $|Q|$ and $\mathbf{p}_{\min}$. In turn, this expectation gives a bound on the (Cesaro-)mixing time of the finite Markov chain induced by the strategy (Levin and Peres, 2017).
- An exploration phase can yield a too imprecise approximation, leading to too suboptimal exploitation phase. However, forgetfulness implies that the effect of any such episode is limited. □

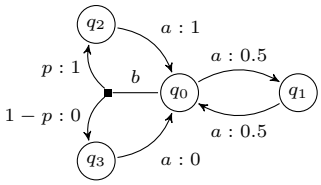


Figure 1: An MDP is depicted for which no finite-memory strategy can achieve better guarantees than almost-sure near-optimality.

These guarantees are optimal for finite-memory strategies as the following example shows.

**Proposition 1.** *Let $\mathcal{M}_p$ be the MDP of Fig. 1 parameterised by probability $p$. For all finite-memory strategies $\sigma$ there exists a probability $p$ such that*

$$\Pr^{q_0}_{\mathcal{M}^\sigma_p}[\rho : \mathbf{MP}(\rho) \geq \mathbf{Val}(\mathcal{M}_p)] < 1.$$

*Proof sketch.* Observe that taking always action $a$ yields a mean payoff of $0.5$, and $b$ yields $p$. Therefore, depending on whether $p < 0.5$, the former or the latter is optimal. Let $\sigma$ be a finite-memory strategy with $n$ states. We perform a case split:

Firstly, assume there exists a run (in $\mathcal{M}_p$ for some $p \in (0,1)$) such that during some $n+1$ consecutive visits of $q_0$, action $a$ is chosen with probability 1. Then, $\sigma$ has revisited $q_0$ with the same memory state twice and in between all choices were deterministic. Thus, $\sigma$ continues looping through these memory states while always choosing action $a$ deterministically. Since the finite prefix has positive probability, the complete run has positive probability in $\mathcal{M}^\sigma_{0.75}$, but it has a mean payoff of $0.5$.

Secondly, we treat the case where in all $n+1$ consecutive visits of $q_0$ on all runs (in e.g. $\mathcal{M}^\sigma_{0.25}$) there is some (constant) positive probability of choosing action $b$. By

the law of large numbers, this implies that action $b$ will be chosen almost surely a nonzero fraction of the time. However, this action will also almost surely contribute an average of $0.25$ reward to the mean payoff and thus the mean payoff is almost surely suboptimal. □

From the above claim it immediately follows that finite-memory strategies exhibit too much regret, motivating PAC learning instead.

**Corollary 1.** *The regret of all finite-memory strategies is at least linear.*

*Proof.* We first introduce the relevant definitions. Let $\mathbf{R}_T = T \cdot \mathbf{Avg}_T$ be the random variable for the total reward in the first $T$ steps. The *regret* of a strategy in an MDP $\mathcal{M}$ is then defined as the expectation of $\mathbf{R}_T - T \cdot \mathbf{Val}(\mathcal{M})$. It is easy to see that a finite-memory strategy whose regret is a sublinear function of $T$ would imply $\Pr^{q_0}_{\mathcal{M}^\sigma}[\rho : \mathbf{MP}(\rho) \geq \mathbf{Val}(\mathcal{M})] = 1$, contradicting Proposition 1. □

# 4 MODAL MARKOV DECISION PROCESSES

From this section on, let $\mathcal{M}$ denote a *modal MDP* (mMDP), which is an MDP (not necessarily strongly connected) together with possibly different lower bound $\underline{T}$ and upper bound $\overline{T}$ on the set of transitions. Since we consider different transitions sets, we introduce the notation $\mathcal{G}_{S,B,T}$ to denote the directed graph with nodes of $S \subseteq Q$ and edges $\{(s,s') \in S \times S : \exists a \in B(s) : (s,a,s') \in T\}$, i.e., restricted to actions available according to $B \subseteq A$ and transitions in $T$ for the given $\underline{T} \subseteq T \subseteq \overline{T}$.

## 4.1 POSSIBLE AND SAFE ECS

Since we do not know the transition relation of the MDP exactly, we cannot determine the ECs of the MDP. Consequently, we cannot optimize the mean payoff within an EC. Instead, we have to define more general versions of this concept, reflecting the modality of the transitions in what may or must be an EC.

**Definition 6.** *A* possible end component *(PEC) of the mMDP $\mathcal{M}$ is a pair $(S,B)$ with $S \subseteq Q$, $B \subseteq A$ s.t.*

- *for all $s \in S$ and all $a \in B(s)$ we have that $\mathrm{supp}(\underline{T},s,a) \subseteq S$ and $\mathrm{supp}(\overline{T},s,a) \cap S \neq \varnothing$,*
- *$\mathcal{G}_{S,B,\overline{T}}$ is strongly connected.*

The former implies that it is *possible* (for some transition set respecting the bounds) to stay within the PEC, the latter that it is possible to see all of the PEC (infinitely often with probability 1). Every EC is a PEC; later on we also use that if $\mathrm{supp}(P(s,a)) = \mathrm{supp}(\overline{T},s,a) \cap S$ for all $s \in S$ and all $a \in B(s)$ then the PEC is an EC.

PECs indicate components of the mMDP where the mean payoff can be optimized if the transition function is favourable. However, if a PEC is no EC, a strategy might leave the PEC during optimization.

**Definition 7.** *A* safe end component *(SEC)* with respect to $s_0 \in S$ *of the mMDP $\mathcal{M}$ is a pair $(S,B)$ with $S \subseteq Q$ and $B \subseteq A$ such that*

- *for all $s \in S$ and all $a \in B(s)$ we have that $\mathrm{supp}(\overline{T},s,a) \subseteq S$,*
- *for all transition relations $T$ with $\underline{T} \subseteq T \subseteq \overline{T}$ and $\mathrm{supp}(T,s) = A(s)$ for all $s \in S$ we have that all states $s \in S$ have a path to $s_0$ in $\mathcal{G}_{S,B,T}$, and*
- *$\mathcal{G}_{S,B,\overline{T}}$ is strongly connected.*

The former implies that it is *certain* (for any transition set respecting the bounds) to stay within the SEC, the latter that it is certain to return to $s_0$ (infinitely often with probability 1), not necessarily visiting all of the SEC. Besides, we keep the possible strong connectivity, hence every SEC is also a PEC. Consequently, SECs provide components where the mean payoff can be optimized "safely", without the risk of leaving the EC of $s_0$.

An EC, PEC or SEC $(S,B)$ is *maximal* (a MEC, MPEC or MSEC) if for all other ECs, PECs or SECs $(S',B')$ with $S \subseteq S'$ and $B(s) \subseteq B'(s)$ for all $s \in S$ it holds $(S,B) = (S',B')$. These maximal components are of particular interest for optimization since they provide the largest components with the given guarantees.

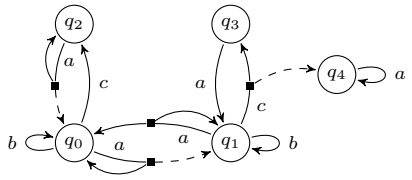


Figure 2: A modal MDP (the may transitions are dashed)

**Example.** Fig. 2 shows a modal MDP. One MPEC is formed by $q_4$, which cannot be left. The other MPEC contains $q_0, q_1, q_2, q_3$ as can be seen by realizing the may transitions to $q_0$ and $q_1$. In contrast, the MSEC of $q_0$ contains only $q_0, q_1$. Indeed, $q_1$ is part of the MSEC even though it may be unreachable from $q_0$; $q_2$ is not in this MSEC since the may transition back to $q_0$ *may not* exist; $q_3$ is not in this MSEC since the may transition to $q_4$ *may* exist, violating the first condition of the definition.

## 4.2 COMPUTATION OF ECS IN MODAL MDPS

In order to optimize within the newly defined ECs, we first present algorithms calculating MPECs (Algo. 1) and MSECs (Algo. 2) in polynomial time. Of course, these algorithms adhere to our assumptions and use only the bounds $\underline{T}$ and $\overline{T}$ on the support of the transition function and thus implicitly the allowed actions $A$.

The idea of the algorithm computing MPECs is to over-approximate all MPECs with larger components and successively refine them until there are only MPECs left. This is very similar to standard algorithms for the calculation of maximal ECs in MDPs, see e.g. (Baier and Katoen, 2008, Algorithm 47). For a component $(S, B)$ in the over-approximation, consider the graph $\mathcal{G}_{S,B,\overline{T}}$.

If this graph is strongly connected, then $(S, B)$ is already a PEC. Otherwise, the graph can be decomposed into strongly connected components (SCCs), and we know that each PEC previously contained in $(S, B)$ has to be contained in one of these SCCs. Thus, we can replace $(S, B)$ with the set of its SCCs. If all components stay unchanged during one iteration of the algorithm, we know that only PECs are left, and these have to be maximal, since we over-approximated them.

In order to give the algorithm for MPECs, we define, for a set $S \subseteq Q$, the *candidate* actions $\underline{B}_S$ for PECs by $\underline{B}_S(s) \stackrel{\text{def}}{=} \{a \in A(s) : \operatorname{supp}(\underline{T}, s, a) \subseteq S \wedge \operatorname{supp}(\overline{T}, s, a) \cap S \neq \varnothing\}$. Intuitively, these actions could belong to a PEC with states $S$.

**Algorithm 1** Calculation of MPECs

**Input:** mMDP $\mathcal{M}$ with transition relation bounds $\underline{T}$, $\overline{T}$
**Ouput:** $M = \{(S, B) : (S, B) \text{ MPEC in } \mathcal{M}\}$

```
M ← {(Q, A)}
repeat
    M' ← ∅
    for all (S, B) ∈ M do
        for all S' ⊆ S : S' is an SCC in G_{S,B,T̄} do
            M' ← M' ∪ {(S', B_{S'})}
    M ← M'
until M stayed unchanged
```

**Theorem 2.** *Algorithm 1 computes the MPECs of a mMDP in polynomial time.*

Considerably more interesting is the algorithm computing MSECs. To this end, we first define, for a set $S \subseteq Q$, the *candidate* actions $\overline{B}_S$ for SECs almost dually as $\overline{B}_S(s) \stackrel{\text{def}}{=} \{a \in A(s) : \operatorname{supp}(\overline{T}, s, a) \subseteq S\}$ for all $s \in S$. Like for PECs, these are all actions that could belong to a SEC with states $S$.

To compute an MSEC w.r.t. $s_0 \in Q$, we analogously over-approximate all SECs and successively refine this approximation. If we have a candidate set $S$ for our MSEC, we only keep those states $S'$ which, using only the candidate actions $\overline{B}_S$ for SECs, definitely have a path to $s_0$, independent of the actual support of the transition function. All SECs have to be contained in these states. In order to compute them, we start with $S' = \{s_0\}$ and then add states $s \in S \setminus S'$ with an action $a \in \overline{B}_S(s)$ which must have a transition to $S'$. These are exactly those actions which cannot possibly stay outside of $S'$, i.e. actions not in $\underline{B}_{Q \setminus S'}(s)$.

If at some point we keep all states and $S$ remains unchanged, we know that we have found a component of the MDP which contains all SECs and fulfils all SEC constraints except for the strong connectedness. Since, by construction, every state in $S$ has a path to $s_0$ we simply need to keep only states reachable from $s_0$ in $\mathcal{G}_{S,\overline{B}_S,\overline{T}}$ to make it a SEC. Since it also contains all SECs, it has to be the unique MSEC w.r.t. $s_0$.

**Algorithm 2** Calculation of MSECs

**Input:** mMDP $\mathcal{M}$ with bounds $\underline{T}$, $\overline{T}$, $s_0 \in Q$
**Ouput:** $(S, B)$ the MSEC with respect to $s_0$

```
S ← Q
repeat
    S' ← {s_0}
    while ∃s ∈ (S \ S') : ∃a ∈ B̄_S(s) \ B_{Q\S'}(s) do
        S' ← S' ∪ {s}
    S ← S'
until S stayed unchanged
S ← {s ∈ S : s is reachable from s_0 in G_{S,B̄_S,T̄}}
(S, B) ← (S, B̄_S)
```

**Theorem 3.** *Algorithm 2 computes the MSEC w.r.t $s_0$ in polynomial time.*

## 5 LEARNING IN MODAL MDPS

In this section, we optimize mean payoff in a modal MDP. Since the MECs of the MDP are not known, the results of Section 3 have to be lifted to the modified notions of MPECs and MSECs of Section 4. Respectively, there are two types of strategies that we want to consider:

1. Either we want to achieve the best mean payoff possible in the MEC in which the strategy ends up in, *not missing any opportunity* in that MEC.
   Secondarily, during the execution of the strategy, we prefer to optimize the mean payoff in the current MEC, taking only minimal risk of leaving it. The motivation for this is that, if possible, we do not skip a lot of promising MECs along the way to the final MEC. Instead, we try to optimize the mean payoff already within them and only move to another MEC if our incomplete knowledge forces us to do so.
2. Or we do not want to take any risk of leaving the current MEC whenever this can be ensured using the partial knowledge.
   This is motivated by the fact that we cannot (almost surely) return to the MEC once we leave it and the lack of information about the rewards out-

side of the current MEC. However, such a conservative strategy might not be able to explore the complete MEC due to avoiding the risk of falling out of it and thus might *miss some opportunities* for a better mean payoff within this MEC. This is a well known issue regarding partially known environments and is best exemplified by the the Canadian traveller's problem (Papadimitriou and Yannakakis, 1991; Nikolova and Karger, 2008).

### 5.1 OPTIMIZATION IN MPECS

We start by considering the first type of strategies. As already stated in Section 4.2, each EC is a PEC and each PEC could constitute an EC. Thus, in order not to miss any opportunity for a good mean payoff in the current MEC, we have to explore the whole current MPEC as it could be a MEC.

To optimize the mean payoff, we employ a strategy similar to the one of Section 3. Again, we use only finite memory to almost surely obtain a mean payoff which is $\varepsilon$-close to the optimal expected mean payoff in the final MEC. Our strategy $\sigma_p$ works in episodes:

- **Explore:** First, $\sigma_p$ chooses actions uniformly at random among the actions *of the current MPEC* $(S, B)$ during $L$ steps to collect statistics allowing it to compute empirical approximations $\widehat{P}$ and $\widehat{R}$ of $P$ and $R$ *in the MEC in which the strategy is at the end of the exploration phase*.
- **Exploit:** Then, $\sigma_p$ follows a (memoryless) expectation-optimal strategy $\tau$ for that MEC in $\widehat{\mathcal{M}} = (Q, q_0, A, \widehat{P}, \widehat{R})$ during $O$ steps. (If it is not in any non-trivial MEC, it can behave arbitrarily.) The strategy $\sigma_p$ then "forgets the learnt model" and restarts from the exploration phase.

Whenever $\sigma_p$ discovers a transition which has not been included in $\underline{T}$, it can update $\underline{T}$ in order to make further calculations of MPECs more precise. However, notice that this is not necessary in order to fulfil the desired guarantees, it only allows in future to explore fewer actions which could lead out of the current MEC and thus it decreases the risk of leaving the MEC.

For a run $\rho \in \mathrm{Runs}^{q_0}(\mathcal{M}^\sigma)$, denote by $\mathrm{Inf}(\rho)$ the set of all states of $Q$ visited infinitely often by $\rho$. Thus, the event $\mathrm{Inf} \subseteq S$ corresponds to all runs eventually staying within $S$. Moreover, for an EC $(S, B)$ in $\mathcal{M}$, let the optimal value in this EC be $\mathbf{Val}(\mathcal{M} \mid S, B) \stackrel{\text{def}}{=} \mathbf{Val}(\mathcal{N})$ where $\mathcal{N} = (S, s_0, B, P|_{S\times A}, R|_{S\times A\times S})$ is the restriction of $\mathcal{M}$ to the EC and $s_0 \in S$ can be arbitrary since this value is the same for all states of an EC. Using this notation, we get the following optimality guarantee:

**Theorem 4.** *For all $\varepsilon \in (0, 1)$, one can compute $L, O \in \mathbb{N}$ s.t. for the resulting finite-state strategy $\sigma_p$ and any MEC $(S, B)$ of $\mathcal{M}$ with $\Pr^{q_0}_{\mathcal{M}^{\sigma_p}}[\mathrm{Inf} \subseteq S] > 0$ we have*

$$\Pr^{q_0}_{\mathcal{M}^{\sigma_p}}[\rho : \mathbf{MP}(\rho) \geq \mathbf{Val}(\mathcal{M}|S, B) - \varepsilon \mid \mathrm{Inf} \subseteq S] = 1$$

*Proof sketch:*

- As $L$ and $O$ of Theorem 1 only depend on $\mathbf{p}_{\min}$, $|Q|$, $|A|$ and $\varepsilon$ and not on the transition relation of $\mathcal{M}$, we can choose them in exactly the same way.
- Consider the moment that $\mathrm{Inf}$ is entered and stayed in forever. The MPEC $(S', B')$ that the strategy computes at that point satisfies $B'|_S = B$. (Otherwise, the MPEC would contain an action leading out of the MEC, which would almost surely be taken in the repetitive exploration and the MEC, in particular this state of $\mathrm{Inf}$, would be left forever.)
- Consequently, $\sigma_p$ actually corresponds to the strategy $\sigma_\varepsilon$ in the MEC $(S, B)$. By our choice of $L$ and $O$, it follows from Theorem 1 that the strategy will obtain an $\varepsilon$-close mean payoff almost surely. □

### 5.2 OPTIMIZATION IN MSECS

Now we consider optimizing mean payoff among "safe" strategies, not leaving the current MEC whenever the partial knowledge allows for that. To this end, we consider strategies which are defined for the maximum supports of the transition function and which never leave the MEC, independent of the actual support.

A *modal strategy* $\sigma$ in a mMDP $\mathcal{M}$ with transition relation bounds $\underline{T}$ and $\overline{T}$ is a strategy in the MDP $\overline{\mathcal{M}} = (Q, q_0, A, \overline{P}, R)$ where $\overline{P}$ is a transition function with $\mathrm{supp}(\overline{P}(q, a)) = \mathrm{supp}(\overline{T}, q, a)$ for all $q \in Q$ and $a \in A(q)$. This means that $\sigma$ is defined for all transition functions compatible with the transition bounds.

A strategy $\sigma$ *is in* $(S, B)$ if for all histories $h$ with $\mathrm{last}(h) \in S$ (and positive probability under $\sigma$) it holds $\mathrm{supp}(\sigma(h)) \subseteq B(\mathrm{last}(h))$. Intuitively, $\sigma$ is in $(S, B)$ if it never leaves it.

Finally, we call a modal strategy $\sigma$ *$s_0$-EC-safe* for some $s_0 \in Q$ if for all transition functions $P'$ compatible with the transition relation bounds $\underline{T}$ and $\overline{T}$ it holds that $\sigma$ is in the MEC containing $s_0$ in $\mathcal{M}' = (Q, s_0, A, P', R)$.

The following lemma establishes the relationship between EC-safe strategies (defined by the desired property) and SECs (effectively computable).

**Lemma 1.** *A modal strategy $\sigma$ is $s_0$-EC-safe iff it is in the MSEC $(S, B)$ w.r.t. $s_0$. Moreover, $\sigma$ is in the EC containing $s_0$ which is the maximal such EC within $(S, B)$.*

Thus in order to optimize mean payoff among $s_0$-EC-safe strategies, we simply have to optimize the mean

payoff in the MSEC w.r.t. $s_0$. In particular, we require the MSEC to be *non-trivial*, i.e. $B(s_0) \neq \varnothing$, since otherwise there does not exist any $s_0$-EC-safe strategy. (For such $s_0$ we first need to reach another state that is in a non-trivial MSEC.) Let the optimal mean-payoff among $s_0$-EC-safe strategies be $\mathbf{sVal}(\mathcal{M}, s_0) \stackrel{\text{def}}{=} \sup_{\tau:\tau \text{ is } s_0\text{-EC-safe}} \mathrm{E}^{s_0}_{\mathcal{M}^\tau}[\mathbf{MP}]$ . We define our finite-memory $s_0$-EC-safe strategy $\sigma_s$, obtaining a mean-payoff which is $\varepsilon$-close to the optimal mean-payoff among $s_0$-EC-safe strategies, using again exploration and exploitation episodes:

- **Explore:** First, $\sigma_s$ chooses actions uniformly at random among the actions *of the MSEC* $(S, B)$ *w.r.t.* $s_0$ during $L$ steps to collect statistics allowing it to compute empirical approximations $\widehat{P}$ and $\widehat{R}$ of $P$ and $R$ *in the EC containing* $s_0$ *which is the maximal such EC within* $(S, B)$.
- **Exploit:** Then, $\sigma_s$ follows a (memoryless) expectation-optimal strategy $\tau$ for that EC in $\widehat{\mathcal{M}} = (Q, q_0, A, \widehat{P}, \widehat{R})$ during $O$ steps. (If no EC is explored, it can execute an arbitrary EC-safe strategy.) The strategy $\sigma_s$ then "forgets the learnt model" and restarts from the exploration phase.

Again, if $\sigma_s$ discovers a new transition, this transition can be added to $\underline{T}$. Either way we get the desired optimality guarantees for $\sigma_s$:

**Theorem 5.** *For all* $\varepsilon \in (0, 1)$*, one can compute* $L, O \in \mathbb{N}$ *such that for the resulting finite-state strategy* $\sigma_s$ *we have* $\Pr^{s_0}_{\mathcal{M}^{\sigma_p}}[\rho : \mathbf{MP}(\rho) \geq \mathbf{sVal}(\mathcal{M}, s_0) - \varepsilon] = 1$.

*Proof sketch:*

- Again, we can choose the $L$ and $O$ as in Theorem 1.
- By Lemma 1, we know that all $s_0$-EC-safe strategies are within the EC $C$ containing $s_0$ which is the maximal such EC within that MSEC. Since $\sigma_s$ is by definition in the MSEC, it is also in $C$.
- Consequently, $\sigma_s$ actually is the strategy $\sigma_\varepsilon$ within $C$. By the choice of $L$ and $O$, Theorem 1 implies the almost sure $\varepsilon$-optimality. □

## 6 EXPERIMENTAL RESULTS

We have implemented the proposed approach and here we illustrate it on several examples. For all simulated runs, exploration phases ran for 1000 steps, exploitation phases for 10000 steps and the number of episodes was set to 3, i.e., 3 exploration and 3 exploitation phases.

In Fig. 3, we can see that 1000 steps of exploration suffice for an approximation of the transition probabilities which was accurate enough to identify the optimal strategy in all simulated runs. The optimal mean payoff of 0.75 is approached by the curves. The slight drops in the curves correspond to the exploration phases where no mean payoff optimization is done. The average reward collected during the simulated prefixes of the runs (with prefix length $3 \cdot (1000 + 10000) = 33000$ steps) amounts to 0.739, corresponding to 98.6 % of the achievable optimum.

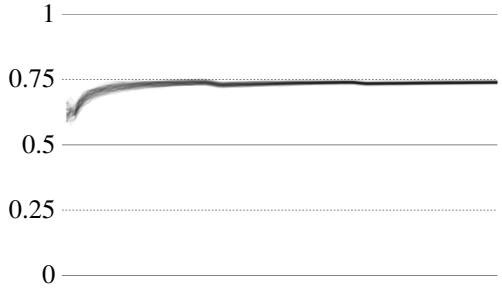


Figure 3: 50 simulation runs in the MDP of Fig. 1 with $p = 0.75$. x-axis: no. of steps, y-axis: average reward.

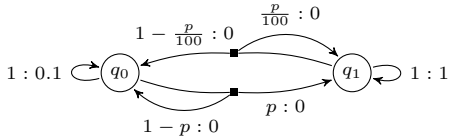


Figure 4: An instantiation of the modal MDP from Fig. 2 (only MSEC is shown) demonstrating the need for $\mathbf{p}_{\min}$.

Fig. 4 shows why the number of steps in the exploration phase has to depend on $\mathbf{p}_{\min}$. Indeed, if this was not the case, $p$ could be chosen small enough such that the probability of never taking the transition to $q_1$ during exploration is arbitrarily close to 1, making it impossible to compute accurate empirical approximations of the transition probabilities. The optimal strategy in Fig. 4 plainly is to try to reach $q_1$ and then always play the right looping transition. However, if $q_1$ is not actually visited during exploration, the strategy will stay in $q_0$ also during optimization, and obtain sub-optimal mean payoff. Moreover, if this transition was a may transition, we would not even know whether it is present at all.

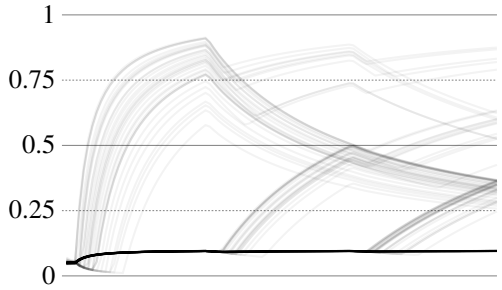


Figure 5: 150 simulation runs in the MDP of Fig. 4 with $p = 0.001$.

In Fig. 5, most of the curves converge towards a mean

payoff of 0.1 since the state $q_1$ is never visited during exploration. Only a small part of the runs ended up in $q_1$ and correctly optimizes the mean payoff. The memory is reset before each exploration phase. Therefore, every run has a new chance to (not) visit $q_1$ during exploration in every episode. This is reflected in the graph by sudden drops or increases of the curves after the exploration phases. If the exploration phase length was chosen according to Section 5.2 instead (taking into account $\mathbf{p}_{\min}$), the collected average reward of 0.262 could be increased to any number $< 1$.

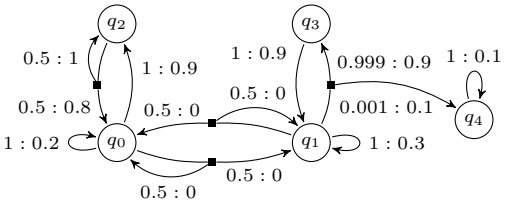


Figure 6: An instantiation of the modal MDP of Fig. 2.

When exploring the MPEC in the mMDP instantiation in Fig. 6, almost all runs will finally end up in $q_4$. Restricting exploration to the MSEC will prevent this behaviour. However, both strategies fail to obtain the optimal mean payoff, which could be collected by always trying to get to $q_2$. This is not a contradiction to Theorem 4 since optimality was only guaranteed conditioned on the event that a run ends up in a particular EC, and for MPEC exploration, all runs will end up in the EC of $q_4$ where the strategy will then be (locally) optimal. Theorem 5 equally does not guarantee global optimality, but only optimality under $q_0$-EC-safe strategies which must never exit the MSEC $q_0, q_1$ (cf. Fig. 2).

When exploring the whole MPEC, runs could already reach $q_4$ during exploration. However, if this does not yet happen, two possibilities arise: depending on how the transition probabilities of the transitions exiting $q_2$ were approximated, the calculated optimizing strategy may try to reach $q_2$ or $q_3$. If $q_3$ is chosen, the run will probably end up in $q_4$ during optimization, explaining the runs with a sudden drop in mean payoff during the optimization phase in Fig. 7. If, however, $q_2$ is chosen, the run will stay within the MPEC for the remaining optimization phase and collect the optimal mean payoff of 0.9 during this phase. In the next exploration phase, it is again possible that $q_2$ or $q_3$ are chosen. Since only 3 episodes were simulated, the part of the runs which always chose $q_2$ and never reached $q_4$ is still significant, as can be seen in the graph. In the long run, however, all runs will end up in $q_4$ and approach a mean payoff of 0.1.

MSEC exploration is straightforward for the instantiation in Fig. 6. The state $q_1$ and the right, self-looping transition are identified as yielding the optimal mean

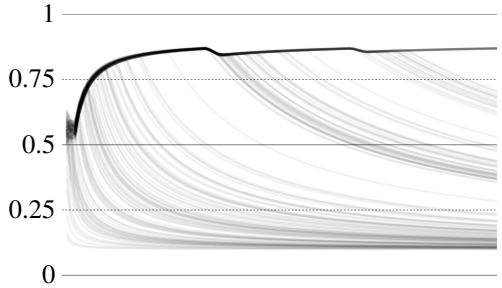


Figure 7: 150 simulation runs in the MDP of Fig. 6 with MPEC exploration.

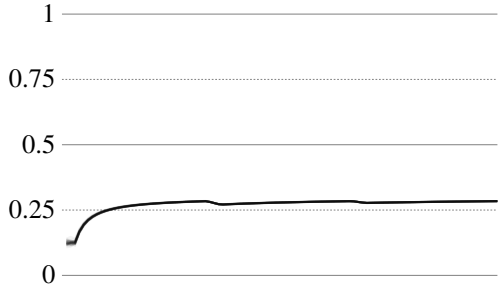


Figure 8: 50 simulation runs in the MDP of Fig. 6 with MSEC exploration.

payoff within the MSEC, and the mean payoff will approach 0.3 (see Fig. 8) which is better than the 0.1 we get from MPEC exploration, but worse than the best possible mean payoff 0.9. After the 33000 simulated steps, the collected average reward of 0.284 is at 94.7 % of the achievable optimum within the MSEC.

## 7 CONCLUSION

We have presented the first algorithms for learning optimal finite-memory strategies in MDPs with mean payoff, both for complete and partial knowledge of the topology (with zero quantitative knowledge). In order to provide a practically more efficient implementation, the next step is to provide smaller number of steps $L$ and $O$ for particular classes of MDP with faster mixing. Alternatively, if the hard guarantees may be relaxed, simulations with low $L$ and $O$ are also possible. In our experiments, they have always proved as either reasonably reliable, or visibly indicating the necessity to increase the numbers, ruling out cases with falsely believed optimality.

**Acknowledgements**

This research was partially funded by the German Research Foundation (DFG) project 383882557 “Statistical Unbounded Verification” (KR 4890/2-1) and the Belgian FWO “SAILor” project (G030020N).

## References


Ashok, P., Kretínský, J., and Weininger, M. (2019). PAC statistical model checking for Markov decision processes and stochastic games. In *CAV (1)*, pages 497–519. Springer.

Auer, P. and Ortner, R. (2006). Logarithmic online regret bounds for undiscounted reinforcement learning. In *NIPS*, pages 49–56. MIT Press.

Baier, C. and Katoen, J.-P. (2008). *Principles of model checking*. MIT Press.

Brafman, R. I. and Tennenholtz, M. (2002). R-MAX - A general polynomial time algorithm for near-optimal reinforcement learning. *J. Mach. Learn. Res.*, pages 213–231.

Brázdil, T., Chatterjee, K., Chmelik, M., Forejt, V., Kretínský, J., Kwiatkowska, M. Z., Parker, D., and Ujma, M. (2014). Verification of Markov decision processes using learning algorithms. In *ATVA*, pages 98–114. Springer.

Daca, P., Henzinger, T. A., Kretínský, J., and Petrov, T. (2016). Faster statistical model checking for unbounded temporal properties. In *TACAS*, pages 112–129. Springer.

Dann, C. and Brunskill, E. (2015). Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2818–2826.

Fu, J. and Topcu, U. (2014). Probably approximately correct MDP learning and control with temporal logic constraints. In *Robotics: Science and Systems*.

Geeraerts, G., Guha, S., and Raskin, J.-F. (2018). Safe and optimal scheduling for hard and soft tasks. In *FSTTCS*, pages 36:1–36:22. Dagstuhl.

Gibney, E. (2016). Google ai algorithm masters ancient game of go. *Nature News*, (7587):445.

Gimbert, H. (2007). Pure stationary optimal strategies in Markov decision processes. In *STACS*, pages 200–211. Springer.

Hahn, E. M., Perez, M., Schewe, S., Somenzi, F., Trivedi, A., and Wojtczak, D. (2019). Omega-regular objectives in model-free reinforcement learning. In *TACAS (1)*, pages 395–412. Springer.

Henriques, D., Martins, J. G., Zuliani, P., Platzer, A., and Clarke, E. M. (2012). Statistical model checking for Markov decision processes. In *QEST*, pages 84–93. IEEE Computer Society.

Jaksch, T., Ortner, R., and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *J. Mach. Learn. Res.*, pages 1563–1600.

Kearns, M. J. and Singh, S. P. (2002). Near-optimal reinforcement learning in polynomial time. *Machine Learning*, (2-3):209–232.

Křetínský, J., Pérez, G. A., and Raskin, J.-F. (2018). Learning-based mean-payoff optimization in an unknown MDP under omega-regular constraints. In *CONCUR*, pages 8:1–8:18. Dagstuhl.

Larsen, K. G. and Thomsen, B. (1988). A modal process logic. In *LICS*, pages 203–210. IEEE Computer Society.

Lassaigne, R. and Peyronnet, S. (2012). Approximate planning and verification for large Markov decision processes. In *SAC*, pages 1314–1319. ACM.

Levin, D. A. and Peres, Y. (2017). *Markov chains and mixing times*. American Mathematical Soc.

Lovász, L. and Winkler, P. (1995). Exact mixing in an unknown Markov chain. *Electr. J. Comb.*

Nikolova, E. and Karger, D. R. (2008). Route planning under uncertainty: The canadian traveller problem. In *AAAI*, pages 969–974.

Papadimitriou, C. H. and Yannakakis, M. (1991). Shortest paths without a map. *Theoretical Computer Science*, (1):127–150.

Propp, J. G. and Wilson, D. B. (1998). How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph. *J. Algorithms*, (2):170–217.

Puterman, M. L. (2005). *Markov Decision Processes*. Wiley-Interscience.

Russell, S. J. and Norvig, P. (2010). *Artificial Intelligence - A Modern Approach (3. internat. ed.)*. Pearson Education.

Schwartz, A. (1993). A reinforcement learning method for maximizing undiscounted rewards. In *ICML*, pages 298–305. Morgan Kaufmann.

Strehl, A. L., Li, L., Wiewiora, E., Langford, J., and Littman, M. L. (2006). PAC model-free reinforcement learning. In *ICML*, pages 881–888. ACM.

Tracol, M. (2009). Fast convergence to state-action frequency polytopes for MDPs. *Opereration Research Letters*, (2):123–126.

Watkins, C. J. C. H. and Dayan, P. (1992). Technical note Q-learning. *Machine Learning*, pages 279–292.