Learning-Based Mean-Payoff Optimization in an Unknown MDP under Omega-Regular Constraints

Jan Křetínský
Technische Universität München, Munich, Germany
jan.kretinsky@in.tum.de
https://orcid.org/0000-0002-8122-2881

Guillermo A. Pérez[1]
Université libre de Bruxelles, Brussels, Belgium
gperezme@ulb.ac.be
https://orcid.org/0000-0002-1200-4952

Jean-François Raskin[2]
Université libre de Bruxelles, Brussels, Belgium
jraskin@ulb.ac.be

Abstract

We formalize the problem of maximizing the mean-payoff value with high probability while satisfying a parity objective in a Markov decision process (MDP) with unknown probabilistic transition function and unknown reward function. Assuming the support of the unknown transition function and a lower bound on the minimal transition probability are known in advance, we show that in MDPs consisting of a single end component, two combinations of guarantees on the parity and mean-payoff objectives can be achieved depending on how much memory one is willing to use. (i) For all ε and γ we can construct an online-learning finite-memory strategy that almost-surely satisfies the parity objective and which achieves an ε-optimal mean payoff with probability at least $1 - \gamma$. (ii) Alternatively, for all ε and γ there exists an online-learning infinite-memory strategy that satisfies the parity objective surely and which achieves an ε-optimal mean payoff with probability at least $1 - \gamma$. We extend the above results to MDPs consisting of more than one end component in a natural way. Finally, we show that the aforementioned guarantees are tight, i.e. there are MDPs for which stronger combinations of the guarantees cannot be ensured.

2012 ACM Subject Classification Theory of computation → Logic and verification, Theory of computation → Reinforcement learning

Keywords and phrases Markov decision processes, Reinforcement learning, Beyond worst case

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2018.8

Related Version A full version is available at `https://arxiv.org/abs/1804.08924`.

Funding This research was funded in part by the Czech Science Foundation grant No. 18-11193S, the German Research Foundation (DFG) project 383882557 "Statistical Unbounded Verification", the ERC Starting grant 279499 "inVEST", the ARC (Fédération Wallonie-Bruxelles) project "Non-Zero Sum Game Graphs: Applications to Reactive Synthesis and Beyond", and the EOS (FNRS-FWO) project 30992574 "Verifying Learning Artificial Intelligence Systems".

[1] G. A. Pérez has been supported by an F.R.S.-FNRS Aspirant fellowship.
[2] J.-F. Raskin is Professeur Francqui de Recherche funded by the Francqui foundation.

29th International Conference on Concurrency Theory (CONCUR 2018).
Editors: Sven Schewe and Lijun Zhang; Article No. 8; pp. 8:1–8:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Reactive synthesis and online reinforcement learning. Reactive systems are systems that maintain a continuous interaction with the environment in which they operate. When designing such systems, we usually face two partially conflicting objectives. First, to ensure a safe execution, we want some basic and critical properties to be enforced by the system no matter how the environment behaves. Second, we want the reactive system to be as efficient as possible given the actual observed behaviour of the environment in which the system is executed. As an illustration, let us consider a robot that needs to explore an unknown environment as efficiently as possible while avoiding any collision. While operating at low speed makes it easier to avoid collisions, it will impair its ability to explore the environment quickly even if the environment is clear of other objects.

There has been, in the past, a large research effort to define mathematical models and algorithms in order to address the two objectives above, but in isolation only. To synthesize safe control strategies, two-player zero-sum games with omega-regular objectives have been proposed [29, 4]. Reinforcement-learning (RL, for short) algorithms for partially-specified Markov decision processes (MDPs) have been proposed (see e.g. [32, 22, 26, 28]) to learn strategies that reach (near-)optimal performance in the actual environment in which the system is executed. In this paper, we want to answer the following question: *How efficient can online-learning techniques be if only correct executions, i.e. executions that satisfy a specified omega-regular objective, are explored during execution?* So, we want to understand how to combine synthesis and RL to construct systems that are safe, yet, at the same time, can adapt their behaviour according to the actual environment in which they execute.

Problem statement. In order to answer in a precise way the question above, we consider a model halfway between the fully-unknown models considered in RL and the full-known models used in verification. To be precise, we consider as input an MDP with rewards whose transition probabilities are not known and whose rewards are discovered on the fly. That is, the input is the support of the unknown transition function of the MDP. This is natural from the point of view of verification since: we may be working with an underspecified system, its qualitative behaviour may have already been observed, or we may not trust all given probability values. As optimization objective on this MDP, we consider the mean-payoff function, and to capture the sure omega-regular constraint we use a parity objective.

Contributions. Given a lower bound $\pi_{\min}$ on the minimal transition probability, we show that, in partially-specified MDPs consisting of a single end component (EC), two combinations of guarantees on the parity and mean-payoff objectives can be achieved. (i) For all ε and γ, we show how to construct a finite-memory strategy which almost-surely satisfies the parity objective and which achieves an ε-optimal mean payoff with probability at least $1-\gamma$ (Prop. 20). (ii) For all ε and γ, we show how to construct an infinite-memory strategy which satisfies the parity objective surely and which achieves an ε-optimal mean payoff with probability at least $1-\gamma$ (Prop. 14). We also extend our results to MDPs consisting of more than one EC in a natural way (Thms. 21 and 16) and study special cases that allow for improved optimality results as in the case of good ECs (Props. 11 and 17). Finally, we show that there are partially-specified MDPs for which stronger combinations of the guarantees cannot be ensured.

Our usage of $\pi_{\min}$ follows [9, 18] where it is argued that it is necessary for the statistical analysis of unbounded-horizon properties and realistic in many scenarios.

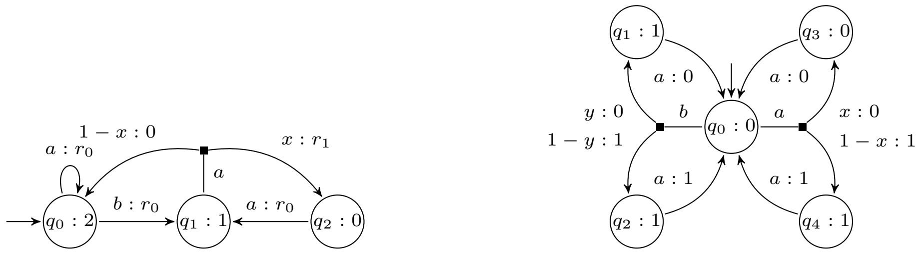

Figure 1 Two automata, representing unknown MDPs, are depicted in the figure. Actions label edges from states (circles) to distributions (squares); a probability-reward pair, edges from distributions to states; an action-reward pair, Dirac transitions; a name-priority pair, states.

Example: almost-sure constraints. Consider the MDP on the right-hand side of Fig. 1 for which we know the support of the transition function but not the probabilities x and y (for simplicity the rewards are assumed to be known). First, note that while there is no surely winning strategy for the parity objective in this MDP, playing action a forever in q_0 guarantees to visit state q_3 infinitely many times with probability one, i.e. this is a strategy that almost-surely wins the parity objective. Clearly, if $x > y$ then it is better to play b for optimizing the mean-payoff, otherwise, it is better to play a. As x and y are unknown, we need to learn estimates $\hat{x}$ and $\hat{y}$ for those values to make a decision. This can be done by playing a and b a number of times from q_0 and by observing how many times we get up and how many times we get down. If $\hat{x} > \hat{y}$, we may choose to play b forever in order to optimize our mean payoff. We then face two difficulties. First, after the learning episode, we may instead observe $\hat{x} < \hat{y}$ while $x \geq y$. This is because we may have been unlucky and observed statistics that differ from the real distribution. Second, playing b always is not an option if we want to satisfy the parity objective with probability 1 (almost surely). In this paper, we give algorithms to overcome these two problems and compute a finite-memory strategy that satisfies the parity objective with probability 1 and is close to the optimal expected mean-payoff value with high probability.

The finite-memory learning strategy produced by our algorithm works as follows in this example. First, it chooses $n \in \mathbb{N}$ large enough so that trying a and b from q_0 as many as n times allows it to learn $\hat{x}$ and $\hat{y}$ such that $|\hat{x} - x| \leq \varepsilon$ and $|\hat{y} - y| \leq \varepsilon$ with probability at least $1 - \gamma$. Then, if $\hat{x} > \hat{y}$ the strategy plays b for K steps and then a once. K is chosen large enough so that the mean payoff of any run will be ε-close to the best obtainable expected mean payoff with probability at least $1 - \gamma$. Furthermore, as a is played infinitely many times, the upper-right state will be visited infinitely many times with probability 1. Hence, the strategy is also almost-surely satisfying the parity objective.

In the sequel we also show that if we allow for learning all along the execution of the strategy then we can get, on this example, the exact optimal value and satisfy the parity objective almost surely. However, to do so, we need infinite memory.

Related works. In [11, 17, 8, 16], we initiated the study of a mathematical model that combines MDPs and two-player zero sum games. With this new model, we provide formal grounds to synthesize strategies that guarantee *both* some minimal performance against any adversary *and* a higher expected performance against a given expected behaviour of the environment, thus essentially combining the two traditional standpoints from games and MDPs. Following this approach, in [1], Almagor et al. study MDPs equipped with a mean-payoff and parity objective. They study the problem of synthesizing a strategy that ensures an expected mean-payoff value that is as large as possible while satisfying a parity objective surely. In [15], Chatterjee and Doyen study how to enforce almost surely a parity objective together with threshold constraint on the expected mean-payoff. See also [10], where mean-payoff MDPs with energy constraints are studied. In all those works, the transition probability and the reward function are *known* in advance. In contrast, we consider the more complex setting in which the reward function is *discovered on the fly during execution time* and the transition probabilities need to be *learned*.

In [19, 33, 21, 2], RL is combined with safety guarantees. In those works, there is a MDP with a set of unsafe states that must be avoided at all cost. This MDP is then restricted to states and actions that are safe and cannot lead to unsafe states. Thereafter, classical RL is exercised. The problem that is considered there is thus very similar to the problem that we study here with the difference that they only consider *safety constraints*. For safety constraints, the reactive synthesis phase and the RL can be entirely decoupled with a two-phase algorithm. A simple two-phase approach cannot be applied to the more general setting of parity objectives. In our more challenging setting, we need to intertwine the learning with the satisfaction of the parity objective in a non trivial way. It is easy to show that reducing parity to safety, as in [7], could lead to learning strategies that are arbitrary far from the optimal value that our learning strategies achieve. In [34], Topcu and Wen study how to learn in a MDP with a discounted-sum (and not mean-payoff) function and liveness constraints expressed as deterministic Büchi automata that must be enforced *almost surely*. Contrary to our setting, they do not consider general omega-regular specifications expressed as parity objectives nor *sure* satisfaction.

Finally, in [9], we apply RL to MDPs where even the topology is unknown. Only $\pi_{\min}$ and, for convenience, the size of the state space is given. There, we optimize the probability to satisfy an omega-regular property; however, no mean payoff is involved.

Structure of the paper. In Sect. 2, we introduce the necessary preliminaries. In Sect. 3, we study online finite and infinite-memory learning strategies for mean-payoff objectives without omega-regular constraints. In Sect. 4, we study strategies for mean-payoff objectives under a parity constraint that must be enforced surely. In Sect. 5, we study strategies for mean-payoff objectives under a parity constraint that must be enforced almost surely.

2 Preliminaries

Let S be a finite set. We denote by $\mathcal{D}(S)$ the set of all *(rational) probabilistic distributions* on S, i.e. the set of all functions $f : S \to \mathbb{Q}_{\geq 0}$ such that $\sum_{s \in S} f(s) = 1$. For sets A and B and functions $g : A \to \mathcal{D}(S)$ and $h : A \times B \to \mathcal{D}(S)$, we write $g(s|a)$ and $h(s|a,b)$ instead of $g(a)(s)$ and $h(a,b)(s)$ respectively. The *support* of a distribution $f \in \mathcal{D}(S)$ is the set $\mathrm{supp}(f) \stackrel{\text{def}}{=} \{s \in S \mid f(s) > 0\}$. The support of a function $g : A \to \mathcal{D}(S)$ is the relation $R \subseteq A \times S$ such that $(a,s) \in R \stackrel{\text{def}}{\iff} g(s|a) > 0$.

2.1 Markov chains

▶ **Definition 1** (Markov chains). A *Markov chain* $\mathcal{C}$ (MC, for short) is a tuple (Q, δ, p, r) where Q is a (potentially countably infinite) set of states, δ is a (probabilistic) transition function $\delta : Q \to \mathcal{D}(Q)$, $p : Q \to \mathbb{N}$ is a priority function, and $r : \mathrm{supp}(\delta) \to [0,1] \cap \mathbb{Q}$ is an (instantaneous) reward function.

A *run* of an MC is an infinite sequence of states $q_0 q_1 \dots \in Q^\omega$ such that $\delta(q_{i+1}|q_i) > 0$ for all $0 \le i$. We denote by $\text{Runs}^{q_0}(\mathcal{C})$ the set of all runs of $\mathcal{C}$ that start with the state q_0.

Consider an initial state q_0. The *probability* of every measurable *event* $\mathcal{A} \subseteq \text{Runs}^{q_0}(\mathcal{C})$ is well-defined [31, 25]. We denote by $\mathbb{P}_{\mathcal{C}}^{q_0}[\mathcal{A}]$ the probability of $\mathcal{A}$; for a measurable function $f : \text{Runs}^{q_0}(\mathcal{C}) \to \mathbb{R}$, we write $\mathbb{E}_{\mathcal{C}}^{q_0}[f]$ for the *expected value* of the function f under the probability measure $\mathbb{P}_{\mathcal{C}}^{q_0}[\cdot]$ (see [23, 25] for a detailed definition of these classical notions).

Parity and mean payoff. Consider a run $\varrho = q_0 q_1 \dots$ of $\mathcal{C}$. We say ϱ *satisfies the parity objective*, written $\varrho \models \textsc{Parity}$, if the minimal priority of states along the run is even. That is to say $\varrho \models \textsc{Parity} \overset{\text{def}}{\iff} \liminf\{p(q_i) \mid i \in \mathbb{N}\}$ is even. In a slight abuse of notation, we sometimes write $\textsc{Parity}$ to refer to the set of all runs of an MC which satisfy the parity objective $\{\varrho \in \text{Runs}^{q_0}(\mathcal{C}) \mid \varrho \models \textsc{Parity}\}$. The latter set of runs is clearly measurable.

The *mean-payoff function* $\underline{\mathbf{MP}}$ is defined for all runs $\varrho = q_0 q_1 \dots$ of $\mathcal{C}$ as follows $\underline{\mathbf{MP}}(\varrho) \overset{\text{def}}{=} \liminf_{j \in \mathbb{N}_{>0}} \frac{1}{j} \sum_{i=0}^{j-1} r(q_i, q_{i+1})$. This function is readily seen to be Borel definable [13], thus also measurable.

2.2 Markov decision processes

▶ **Definition 2** (Markov decision processes). A *(finite discrete-time) Markov decision process* $\mathcal{M}$ (MDP, for short) is a tuple $(Q, A, \alpha, \delta, p, r)$ where Q is a finite set of states, A a finite set of actions, $\alpha : Q \to \mathcal{P}(A)$ a function that assigns to q its set of available actions, $\delta : Q \times A \to \mathcal{D}(Q)$ a (partial probabilistic) transition function with $\delta(q, a)$ defined for all $q \in Q$ and all $a \in \alpha(q)$, $p : Q \to \mathbb{N}$ a priority function, and $r : \text{supp}(\delta) \to [0, 1] \cap \mathbb{Q}$ a reward function. We make the assumption that $\alpha(q) \neq \varnothing$ for all $q \in Q$, i.e. there are no deadlocks.

A *history* h in an MDP is a finite state-reward-action sequence that ends in a state and respects α, δ, and r, i.e. if $h = q_0 a_0 x_0 \dots a_{k-1} x_{k-1} q_k$ then $a_i \in \alpha(q_i)$, $\delta(q_{i+1}|q_i, a_i) > 0$, and $r(q_i, a_i, q_{i+1})$, for all $0 \le i < k$. We write $\text{last}(h)$ to denote the state q_k. For two histories h, h', we write $h < h'$ if h is a *proper prefix* of h'.

▶ **Definition 3** (Strategies). A *strategy* σ in an MDP $\mathcal{M} = (Q, A, \alpha, \delta, p, r)$ is a function $\sigma : (Q \cdot A \cdot \mathbb{Q})^* Q \to \mathcal{D}(A)$ such that $\sigma(a|h) > 0 \implies a \in \alpha(\text{last}(h))$.

We write that a strategy σ is *memoryless* if $\sigma(h) = \sigma(h')$ whenever $\text{last}(h) = \text{last}(h')$; *deterministic* if for all histories h the distribution $\sigma(h)$ is Dirac.

Throughout this work we will speak of *steps*, *episodes*, and *following strategies*. We write that σ *follows* τ *(from the history $h = q_0 a_0 x_0 \dots q_k$) during n steps* if for all $h' = q'_0 a'_0 x'_0 \dots q'_\ell$, such that $h < h'$ and $\ell \le k + n$, we have that $\sigma(h') = \tau(h')$. An episode is simply a finite sequence of consecutive steps, i.e. a finite infix of the history, during which one or more strategies may have been sequentially followed.

A *stochastic Mealy machine* $\mathcal{T}$ is a tuple (M, m_0, f_u, f_o) where M is a (potentially countably infinite) set of memory elements, $m_0 \in M$ is the initial memory element, $f_u : M \times Q \times \mathbb{Q} \to M$ is an update function, and $f_o : M \times Q \to \mathcal{D}(A)$ is an output function. The machine $\mathcal{T}$ is said to implement a strategy σ if for all histories $h = q_0 a_0 x_0 \dots a_{k-1} x_{k-1} q_k$ we have $\sigma(h) = f_o(m_k, q_k)$, where m_k is inductively defined as $m_i = f_u(m_{i-1}, q_{i-1}, x_{i-1})$ for all $i \ge 1$. It is easy to see that any strategy can be implemented by such a machine. A strategy σ is said to have *finite memory* if there exists a stochastic Mealy machine that implements it and such that its set M of memory elements is finite.

A (possibly infinite) state-action sequence $h = q_0 a_0 x_0 q_1 a_1 x_1 \dots$ is *consistent with strategy* σ if $\sigma(a_i | q_0 a_0 x_0 \dots a_{i-1} x_{i-1} q_i) > 0$ for all $i \ge 0$.

From MDPs to MCs. The MDP $\mathcal{M}$ and a strategy σ implemented by the stochastic Mealy machine (M, m_0, f_u, f_0) induce the MC $\mathcal{M}^\sigma = (Q', \delta', p', r')$ where $Q' = (Q \times M \times A) \cup (Q \times M)$; $\delta'(\langle q', m', a' \rangle | s) = f_o(a'|m, q) \cdot \delta(q'|q, a')$ for any $s \in \{\langle q, m, a \rangle, \langle q, m \rangle\}$ and $a' \in \alpha(q)$ with $(q, a', q') \in \operatorname{supp}(\delta)$ and $m' = f_u(m, q, r(q, a', q'))$; $p'(\langle q, m, a \rangle) = p'(\langle q, m \rangle) = p(q)$; and $r'(s, \langle q', m', a' \rangle) = r(q, a, q')$ for any $s \in \{\langle q, m, a \rangle, \langle q, m \rangle\}$. For convenience, we write $\mathbb{P}^{q_0}_{\mathcal{M}^\sigma}[\cdot]$ instead of $\mathbb{P}^{\langle q_0, m_0 \rangle}_{\mathcal{M}^\sigma}[\cdot]$.

A strategy σ is said to be *unichain* if $\mathcal{M}^\sigma$ has a single recurrent class, i.e. a single bottom strongly-connected component (BSCC).

End components. Consider a pair (S, β) where $S \subseteq Q$ and $\beta : S \to \mathcal{P}(A)$ gives a subset of actions allowed per state (i.e. $\beta(q) \subseteq \alpha(q)$ for all $q \in S$). Let $\mathcal{G}_{(S,\beta)}$ be the directed graph (S, E) where E is the set of all pairs $(q, q') \in S \times S$ such that $\delta(q'|q, a) > 0$ for some $a \in \beta(q)$. We say (S, β) is an *end component* (EC) if the following hold: if $a \in \beta(s)$, for $(s, a) \in S \times A$, then $\operatorname{supp}(\delta(s, a)) \subseteq S$; and the graph $\mathcal{G}_{(S,\beta)}$ is strongly connected. Furthermore, we say the EC (S, β) is *good (for the parity objective)* (a GEC, for short) if the minimal priority over all states from S is even; *weakly good* if it contains a GEC.

For ECs (S, β) and (S', β'), let us denote by $(S, \beta) \subseteq (S', \beta')$ the fact that $S \subseteq S'$ and $\beta(s) \subseteq \beta'(s)$ for all $s \in S$. We denote by $\mathrm{MEC}_{\mathcal{M}}$ the set of all maximal ECs (MECs) in $\mathcal{M}$ with respect to $\subseteq$. It is easy to see that for all $(S, \cdot), (S', \cdot) \in \mathrm{MEC}_{\mathcal{M}}$ we have that $S \cap S' = \varnothing$, i.e. every state belongs to at most one MEC.

Model learning and robust strategies. In this work we will "approximate" the stochastic dynamics of an unknown EC in an MDP. Below, we formalize what we mean by approximation.

▶ **Definition 4** (Approximating distributions). Let $\mathcal{M} = (Q, A, \alpha, \delta, p, r)$ be an MDP, (S, β) an EC, and $\varepsilon \in (0, 1)$. We say δ' is ε-close to δ in (S, β), denoted $\delta' \sim^{\varepsilon}_{(S,\beta)} \delta$, if $|\delta'(q'|q, a) - \delta(q'|q, a)| \leq \varepsilon$ for all $q, q' \in S$ and all $a \in \beta(q)$. If the inequality holds for all $q, q' \in Q$ and all $a \in \alpha(q)$, then we write $\delta' \sim^{\varepsilon} \delta$.

A strategy σ is said to be *(uniformly) expectation-optimal* if for all $q_0 \in Q$ we have $\mathbb{E}^{q_0}_{\mathcal{M}^\sigma}[\underline{\mathbf{MP}}] = \sup_\tau \mathbb{E}^{q_0}_{\mathcal{M}^\tau}[\underline{\mathbf{MP}}]$. The following result captures the idea that some expectation-optimal strategies for MDPs whose transition function have the same support are "robust". That is, when used to play in another MDP with the same support and close transition functions, they achieve near-optimal expectation.

▶ **Lemma 5** (Follows from [27, Theorem 6] and [14, Theorem 5]). *Consider values $\varepsilon, \eta_\varepsilon \in (0, 1)$ such that $\eta_\varepsilon \leq \frac{\varepsilon \cdot \pi_{\min}}{24|Q|}$; a transition function δ' such that $\operatorname{supp}(\delta) = \operatorname{supp}(\delta')$ and $\delta \sim^{\eta_\varepsilon} \delta'$ where $\pi_{\min}$ is the minimal nonzero probability value from δ and δ'; and a reward function r' such that $\max\{|r(q, a, q') - r'(q, a, q')| : (q, a, q') \in \operatorname{supp}(\delta)\} \leq \frac{\varepsilon}{4}$. For all memoryless deterministic expectation-optimal strategies σ in $(Q, A, \alpha, \delta', p, r')$, for all $q_0 \in Q$, it holds that $|\mathbb{E}^{q_0}_{\mathcal{M}^\sigma}[\underline{\mathbf{MP}}] - \sup_\tau \mathbb{E}^{q_0}_{\mathcal{M}^\tau}[\underline{\mathbf{MP}}]| \leq \varepsilon$.*

We say a strategy σ such as the one in the result above is *ε-robust-optimal* (with respect to the expected mean payoff).

2.3 Automata as proto-MDPs

We study MDPs with unknown transition and reward functions. It is therefore convenient to abstract those values and work with *automata*.

▶ **Definition 6** (Automata). A *(finite-state parity) automaton* $\mathcal{A}$ is a tuple (Q, A, T, p) where Q is a finite set of states, A is a finite alphabet of actions, $T \subseteq Q \times A \times Q$ is a transition relation, and $p : Q \to \mathbb{N}$ is a priority function. We make the assumption that for all $q \in Q$ we have $(q, a, q') \in T$ for some $(a, q') \in A \times Q$.

A transition function $\delta : Q \times A \to \mathcal{D}(Q)$ is then said to be *compatible* with $\mathcal{A}$ if $\forall (q, a) \in Q \times A : \mathrm{supp}\,(\delta(q, a)) = \{q' \mid T(q, a, q')\}$. For a transition function δ compatible with $\mathcal{A}$ and a reward function $r : T \to [0, 1] \cap \mathbb{Q}$, we denote by $\mathcal{A}_{\delta,r}$ the MDP $(Q, A, \alpha_T, \delta, p, r)$ where $a \in \alpha_T(q) \overset{\text{def}}{\iff} \exists (q, a, q') \in T$. It is easy to see that the sets of ECs of MDPs $(Q, A, \alpha_T, \delta, p, r)$ and $(Q, A, \alpha_T, \delta', p, r')$ coincide for all δ' compatible with $\mathcal{A}$ and all reward functions r'. Hence, we will sometimes speak of the ECs of an automaton.

Example: sure-constraints. Consider the (variable-labelled-)automaton on the left-hand side of Fig. 1. Note that playing a forever surely wins the parity objective from everywhere but it may not be optimal for the expected mean payoff. To play optimally, we need to learn about the values r_1, r_2, and x. Assume that we play for n steps a and b uniformly at random when in state q_0. This will probably allows us to reach q_1 and q_2 a number of times, and so to learn r_0 and r_1, and compute an estimation $\hat{x}$ of x. If $\hat{x} \cdot r_1 > r_0$, we may want to conclude that the optimal strategy is to always play b from q_0. But we face here two major difficulties. First, after the learning episode of n steps, we can observe $\hat{x} \cdot r_1 > r_0$ while $x \cdot r_1 \leq r_0$, this is because we may have been unlucky and observed statistics that differ from the real distribution. Second, playing b always is not an option if we want to surely satisfy the parity objective. In this paper, we give algorithms to overcome the two problems. In our example, the strategy constructed by our algorithm will do the following: given $\varepsilon, \gamma \in (0, 1)$, choose $n \in \mathbb{N}$ large enough, learn $\hat{x}$ such that $|\hat{x} - x| \leq \varepsilon$ with probability more than $1 - \gamma$, then if $\hat{x} \cdot r_1 \leq r_0$, play a forever. Otherwise, keep playing b for longer and longer episodes. If during one of these episodes, the state q_2 is not visited (i.e. the parity objective is endangered as the minimal priority seen during the episode is odd) switch to playing a forever.

Transition-probability lower bound. Let $\pi_{\min} \in [0, 1] \cap \mathbb{Q}$ be a *transition-probability lower bound*. We say that δ is *compatible* with $\pi_{\min}$ if for all $(q, a, q') \in Q \times A \times Q$ we have that: either $\delta(q'|q, a) \geq \pi_{\min}$ or $\delta(q'|q, a) = 0$.

3 Learning for MP: the Unconstrained Case

In this section, we focus on the design of optimal learning strategies for the mean-payoff function in the unconstrained single-end-component case. That is, we have an unknown strongly connected MDP with no parity objective.

We consider, in turn, learning strategies that use finite and infinite memory. Whereas classical RL algorithms focus on achieving an optimal expected value (see, e.g., [32]; cf. [6]), we prove here that a stronger result is achievable: one can ensure – using finite memory only – outcomes that are close to the best expected value with high probability. Further, with infinite memory the optimal outcomes can be ensured with probability 1. In both cases, we argue that our results are tight.

For the rest of this section, let us fix an automaton $\mathcal{A} = (Q, A, T, p)$ such that (Q, α_T) is an EC, and some $\pi_{\min} \in (0, 1]$.

Yardstick. Let δ be a transition function compatible with $\mathcal{A}$ and $\pi_{\min}$, and r be a reward function. The optimal expected mean-payoff value that is achievable in the unique EC (Q, α_T) is defined as $\mathbf{Val}(Q, \alpha_T) \stackrel{\text{def}}{=} \sup_\sigma \mathbb{E}^{q_0}_{\mathcal{A}^\sigma_{\delta,r}}[\underline{\mathbf{MP}}]$ for any $q_0 \in Q$. Indeed, it is well known that this value is the same for all states in the same EC.

Note that this value can always be obtained by a memoryless deterministic [20] and unichain [11] expectation-optimal strategy when δ and r are known. We will use this value as a yardstick for measuring the performance of the learning strategies we describe below.

Model learning. Our strategies learn approximate models of δ and r to be able to compute near-optimal strategies. To obtain those models, we use an approach based on ideas from probably approximately correct (PAC) learning. Namely, we will execute a random exploration of the MDP for some number of steps and obtain an empirical estimation of its stochastic dynamics, see e.g. [30]. We say that a memoryless strategy λ is a *(uniform random) exploration strategy* for a function $\beta : Q \to \mathcal{P}(A)$ if $\lambda(a|q) = 1/|\beta(q)|$ for all $q \in Q, a \in \alpha(q)$ such that $a \in \beta(q)$ and $\lambda(a|q) = 0$ otherwise. Each time the random exploration enters a state q and chooses an action a, we say that it performs an experiment on (q, a), and if the state reached is q' then we say that the result of the experiment is q'. Furthermore, the value $r(q, a, q')$ is then known to us. To learn an approximation δ' of the transition function δ, and to learn r, the learning strategy remembers statistics about such experiments. If the random exploration strategy is executed long enough then it collects sufficiently many experiment results to accurately approximate the transition function δ and the exact reward function r with high probability.

The next lemma gives us a bound on the number of $|Q|$-step episodes for which we need to exercise such a strategy to obtain the desired approximation with at least some given probability. It can be proved via a simple application of Hoeffding's inequality.

▶ **Lemma 7.** *For all ECs (S, β) and all $\varepsilon, \gamma \in (0, 1)$ one can compute $n \in \mathbb{N}$ (exponential in $|Q|$ and polynomial in $|A|$, $\pi_{\min}^{-1}$, $\ln(\gamma^{-1})$, and ε^{-1}) such that following an exploration strategy for β during n (potentially non-consecutive) episodes of $|Q|$-steps suffices to collect enough information to be able to compute a transition function δ' such that $\mathbb{P}\left[\delta' \sim^{\varepsilon}_{(S,\beta)} \delta\right] \geq 1 - \gamma$.*

3.1 Finite memory

We now present a family of finite memory strategies σ_{fin} that force, given any $\varepsilon, \gamma \in (0, 1)$, outcomes with a mean payoff that is ε-close to the optimal expected value with probability higher than $1 - \gamma$. The strategy σ_{fin} is defined as follows.

1. First, σ_{fin} follows the model-learning strategy above for L steps, according to Lemma 7, in order to obtain an approximation δ' of δ such that $\delta' \sim^\eta \delta$ with probability at least $1 - \gamma$. A reward function r' is also constructed from the observed rewards.
2. Then, σ_{fin} follows a memoryless deterministic expectation-optimal strategy τ for $\mathcal{A}_{\delta',r'}$.

The following result tells us that if the learning phase is sufficiently long, then we can obtain, with σ_{fin}, a near-optimal mean payoff with high probability.

▶ **Proposition 8.** *For all $\varepsilon, \gamma \in (0, 1)$, one can compute $L \in \mathbb{N}$ such that for the resulting finite memory strategy σ_{fin}, for all $q_0 \in Q$, for all δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and for all reward functions r, we have $\mathbb{P}^{q_0}_{\mathcal{A}^{\sigma_{\text{fin}}}_{\delta,r}}[\varrho : \underline{\mathbf{MP}}(\varrho) \geq \mathbf{Val}(Q, \alpha_T) - \varepsilon] \geq 1 - \gamma$.*

Proof. We will make use of Lemma 5. For that purpose, let $\eta = \min\{\pi_{\min}, \eta_\varepsilon\}$ where η_ε is as in the statement of the lemma. Next, we set $L = |Q|n$ where n is as dictated by Lemma 7 using η and γ. By Lemma 7, with probability at least $1 - \gamma$ our approximation δ' is such that $\delta' \sim^\eta \delta$. Since $\eta \leq \pi_{\min}$, it follows that $\mathrm{supp}(\delta) = \mathrm{supp}(\delta')$ and we now have learned r, again with probability $1 - \gamma$. Finally, since $\eta \leq \eta_\varepsilon$, Lemma 5 implies the desired result. ◀

▶ Remark (Finite-memory implementability). Note that σ_{fin}, as we described it previously, is not immediately seen to be a computable finite stochastic Mealy machine. Let us consider all possible histories of length L. Observe that this set is not finite because of the unknown rewards which can range over arbitrary rational numbers in $[0, 1]$. However, we can finitize the set by focusing only on rewards of bounded representation size by imposing an upper-bound on the bitsize of their representation (truncating the rest off observed rewards) while still satisfying the hypotheses of Lemma 5. Now, for all such histories we can compute an approximation δ' of δ and an approximation r' of the observed reward function r. Using that information, the required finite-memory expectation-optimal strategy τ can be computed. We encode these (finitely many) strategies into the machine implementing σ_{fin} so that it only has to choose which one to follow forever after the (finite) learning phase has ended. Hence, one can indeed construct a finite-memory strategy realizing the described strategy.

Optimality. The following tells us that we cannot do better with finite memory strategies.

▶ **Proposition 9.** *Let $\mathcal{A}$ be the single-EC automaton on the right-hand side of Fig. 1 and $\pi_{\min} \in (0, 1]$. For all $\varepsilon, \gamma \in (0, 1)$, the following two statements hold.*

- *For all finite memory strategies σ, there exist δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and a reward function r, such that $\mathbb{P}^{q_0}_{\mathcal{A}^\sigma_{\delta,r}}\left[\varrho : \underline{\mathbf{MP}}(\varrho) \geq \mathbf{Val}(Q, \alpha_T) - \varepsilon\right] < 1$.*
- *For all finite memory strategies σ, there exist δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and a reward function r such that $\mathbb{P}^{q_0}_{\mathcal{A}^\sigma_{\delta,r}}\left[\varrho : \underline{\mathbf{MP}}(\varrho) < \mathbf{Val}(Q, \alpha_T)\right] \geq \gamma$.*

Proof sketch. With a finite-memory strategy we cannot satisfy a stronger guarantee than being ε-optimal with probability at least $1 - \gamma$ in this example. Indeed, as we can only use finite memory, we can only learn imprecise models of δ and r. That is, we will always have a non-zero probability to have approximated x or y arbitrarily far from their actual values. It should then be clear that neither optimality with high probability nor almost-sure ε-optimality can be achieved. ◀

3.2 Infinite memory

While we have shown that probably approximately optimal is the best that can be obtained with finite memory learning strategies, we now establish that with infinite memory, one can guarantee almost sure optimality.

To this end, we define a strategy σ_∞ which operates in episodes consisting of two phases: learning and optimization. In episode $i \in \mathbb{N}$, the strategy does the following.

1. It first follows an exploration strategy λ for α_T during L_i steps, there exist models δ_i and r_i based on the experiments obtained throughout the $\sum_{j=0}^{i} L_j$ steps during which λ has been followed so far.
2. Then, σ_∞ follows a unichain memoryless deterministic expectation-optimal strategy $\sigma^{\delta_i}_{\mathrm{MP}}$ for $\mathcal{A}_{\delta_i,r_i}$ during O_i steps.

One can then argue that σ_∞ can be instantiated so that in every episode the finite average obtained so far gets ever close to $\mathbf{Val}(Q, \alpha_T)$ with ever higher probability. This is achieved by choosing the L_i as an increasing sequence so that the approximations δ_i get ever better with ever higher probability. Then, the O_i are chosen so as to compensate for the past history, for the time before the induced MC reaches its limit distribution, and for the future number of steps that will be spent learning in the next episode. The latter then allows us to use the Borel-Cantelli lemma to show that in the unknown EC we can obtain its value almost surely.

▶ **Proposition 10.** *One can compute a sequence $(L_i, O_i)_{i\in\mathbb{N}}$ such that $L_i \geq |Q|$ for all $i \in \mathbb{N}$; additionally the resulting strategy σ_∞ is such that for all $q_0 \in Q$, for all δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and for all reward functions r, we have $\mathbb{P}^{q_0}_{\mathcal{A}^{\sigma_\infty}_{\delta,r}}[\varrho : \underline{\mathbf{MP}}(\varrho) \geq \mathbf{Val}(Q, \alpha_T)] = 1$.*

Optimality. Note that σ_∞ is optimal since it obtains with probability 1 the best value that can be obtained when the MDP is fully known, i.e. when δ and r are known in advance.

4 Learning for MP under a Sure Parity Constraint

We show here how to design learning strategies that obtain near-optimal mean-payoff values while ensuring that all runs satisfy a given parity objective with certainty.

First, we note that all such learning strategies must avoid entering states q from which there is no strategy to enforce the parity objective with certainty. Hence, we make the hypothesis that all such states have been removed from the automaton $\mathcal{A}$, and so we assume that for all $q_0 \in Q$ there exists a strategy σ_{par} such that for all functions δ compatible with $\mathcal{A}$, for all reward functions r, and for all $\varrho \in \text{Runs}^{q_0}(\mathcal{A}^\sigma_{\delta,r})$, we have $\varrho \models \textsc{Parity}$. It is worth noting that, in fact, there exists a memoryless deterministic strategy such that the condition holds for all $q_0 \in Q$ [4, 3]. Notice the swapping of the quantifiers over the initial states and the strategy, this is why we say it is *uniformly winning (for the parity objective)*. The set of states to be removed, along with a uniformly winning strategy, can be computed in quasi-polynomial time [12]. We say that an automaton with no states from which there is no winning strategy is *surely good*.

We study the design of learning strategies for mean-payoff optimization under *sure* parity constraints for increasingly complex cases.

4.1 The case of a single good EC

Consider a surely-good automaton $\mathcal{A} = (Q, A, T, p)$ such that (Q, α_T) is a GEC, i.e. the minimal priority of a state in the EC is even, and some $\pi_{\min} \in (0, 1]$.

Yardstick. For this case, we use as yardstick the optimal expected mean-payoff value: $\mathbf{Val}(Q, \alpha_T) = \sup_\sigma \mathbb{E}^{q_0}_{\mathcal{A}^\sigma_{\delta,r}}[\underline{\mathbf{MP}}]$.

Learning strategy. We show here that it is possible to obtain an optimal mean-payoff with high probability. Note that our solution extends a result given by Almagor et al. [1] for *known* MDPs. The main idea behind our solution is to use the strategy σ_∞ from Proposition 10 in a controlled way: we verify that during all successive learning and optimization episodes, the minimal parity value that is visited is even. If during some episode, this is not the case, then we resort to a strategy σ_{par} that enforces the parity objective with certainty. Such σ_{par} is guaranteed to exist as $\mathcal{A}$ is surely good.

▶ **Proposition 11.** *For all $\gamma \in (0, 1)$, there exists a strategy σ such that for all $q_0 \in Q$, for all δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and for all reward functions r, we have $\varrho \models \textsc{Parity}$ for all $\varrho \in \text{Runs}^{q_0}(\mathcal{A}^\sigma_{\delta,r})$ and $\mathbb{P}^{q_0}_{\mathcal{A}^\sigma_{\delta,r}}[\varrho : \underline{\mathbf{MP}}(\varrho) \geq \mathbf{Val}(Q, \alpha_T)] \geq 1 - \gamma$.*

Proof sketch. We modify σ_∞ so as to "give up" on optimizing the mean payoff if the minimal even priority has not been seen during a long sequence of episodes. This will guarantee that the measure of runs which give up on the mean-payoff optimization is at most γ.

First, recall that we can instantiate σ_∞ so that $L_i \geq |Q|$ for all $i \in \mathbb{N}$. Hence, with some probability $\zeta > 0$, during every learning phase, we visit a state with even minimal priority. We can then find a sequence $n_1, n_2, \dots \in \mathbb{N}^\omega$ of natural numbers such that $\prod_{j=i}^{\infty}(1-\zeta^{n_j}) \geq 1-\gamma$, for some $i \in \mathbb{N}$. Given this sequence, we apply the following monitoring. If for $\ell \in \mathbb{N}$ we write $N_\ell \stackrel{\text{def}}{=} \sum_{k=1}^{\ell-1} n_k$, then at the end of the ℓ-th episode we verify that during some learning phase from $L_{N_\ell}, L_{N_\ell+1}, \dots, L_{N_\ell+n_\ell}$ we have visited a state with minimal even priority, otherwise we switch to a parity-winning strategy forever. ◀

Optimality. The following proposition tells us that the guarantees from Proposition 11 are indeed optimal w.r.t. our chosen yardstick.

▶ **Proposition 12.** *Let $\mathcal{A}$ be the single-GEC automaton on the left-hand side of Fig. 1 and $\pi_{\min} \in (0,1]$. For all parity-winning strategies σ, there exist δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and a reward function r, such that $\mathbb{P}^{q_0}_{\mathcal{A}^\sigma_{\delta,r}}[\varrho : \underline{\mathbf{MP}}(\varrho) \geq \mathbf{Val}(Q, \alpha_T)] < 1$.*

Proof sketch. Consider a reward function such that $r_0 = 0$ and $r_1 = 1$ and an arbitrary δ. It is easy to see that $\mathbf{Val}(Q, \alpha_T) = 1$. However, any strategy that ensures the parity objective is satisfied surely must be such that, with probability $\gamma > 0$, it switches to follow a strategy $q_2 \mapsto (a \mapsto 1)$ forever. Hence, with probability at least γ its mean-payoff is sub-optimal. ◀

4.2 The case of a single EC

We now turn to the case where the surely-good automaton $\mathcal{A} = (Q, A, T, p)$ consists of a unique, not necessarily good, EC (Q, α_T). Let us also fix some $\pi_{\min} \in (0,1]$.

An important observation regarding single-end-component MDPs that are surely good is that they contain at least one GEC as stated in the following lemma.

▶ **Lemma 13.** *For all surely-good automata $\mathcal{A} = (Q, A, T, p)$ such that (Q, α_T) is an EC there exists $(S, \beta) \subseteq (Q, \alpha_T)$ such that (S, β) is a GEC in $\mathcal{A}_{\delta,r}$ for all δ compatible with $\mathcal{A}$ and all reward functions r, i.e. (Q, α_T) is weakly good.*

Yardstick. Let δ and r be fixed in the single EC, our yardstick for this case is defined as follows: $\mathbf{sVal}(Q, \alpha_T) \stackrel{\text{def}}{=} \max_{q \in Q} \sup \left\{ \mathbb{E}^q_{\mathcal{A}^\sigma_{\delta,r}}[\underline{\mathbf{MP}}] \,\middle|\, \sigma \text{ is a parity-winning strategy} \right\}$. That is $\mathbf{sVal}(Q, \alpha_T)$ is the best MP expectation value that can be obtained from a state $q \in Q$ with a parity-winning strategy. It is remarkable to note that we take the maximal value over all states in Q. As noted by Almagor et al. [1], this value is not always achievable even when δ and r are a priori known, but it can be approached arbitrarily close.

Learning strategy. The following proposition tells us that we can obtain a value close to $\mathbf{sVal}(Q, \alpha_T)$ with arbitrarily high probability while satisfying the parity objective surely.

▶ **Proposition 14.** *For all $\varepsilon, \gamma \in (0,1)$ there exists a strategy σ such that for all $q_0 \in Q$, for all δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and for all reward functions r, we have $\varrho \models$ PARITY for all $\varrho \in \mathrm{Runs}^{q_0}(\mathcal{A}^\sigma_{\delta,r})$ and $\mathbb{P}^{q_0}_{\mathcal{A}^\sigma_{\delta,r}}[\varrho : \underline{\mathbf{MP}}(\varrho) \geq \mathbf{sVal}(Q, \alpha_T) - \varepsilon] \geq 1 - \gamma$.*

Proof sketch. We define a strategy σ as follows. Let $\eta = \min\{\pi_{\min}, \eta_{\varepsilon/2}\}$ for $\eta_{\varepsilon/2}$ as defined for Lemma 5. The strategy σ plays as follows.

1. It first computes δ' such that $\delta' \sim^\eta \delta$ with probability at least $1 - \gamma/4$ and a reward function r' by following an exploration strategy λ for α_T during J_0 steps (see Lemma 7).

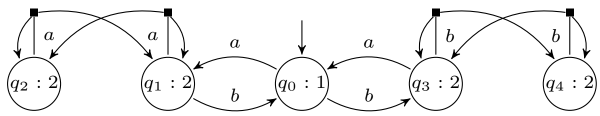

Figure 2 An automaton for which it is impossible to learn to obtain near-optimal mean-payoff almost surely or optimal mean-payoff with high probability, while satisfying the parity objective. For clarity, probability and reward placeholders have been omitted.

2. It then selects a contained maximal good EC (MGEC) with maximal expected mean-payoff value (see Lemma 13) and tries to reach it with probability at least $1 - \gamma/4$ by following λ during J_1 steps.
3. Finally, if the component is reached, it follows a strategy τ as in Proposition 11 with $\gamma/4$ from then onward.

If the learning "fails" or if the component is not reached, the strategy reverts to following a winning strategy forever. (A failed learning phase is one in which the approximated distribution function does not have T as its support.) ◀

Optimality. The following states that we cannot improve on the result of Proposition 14.

▶ **Proposition 15.** *Let $\mathcal{A}$ be the single-EC automaton in Fig. 2 and $\pi_{\min} \in (0, 1]$. For all $\varepsilon, \gamma \in (0, 1)$, the two following statements hold.*

- *For all strategies σ, there exist δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and a reward function r, such that $\mathbb{P}^{q_0}_{\mathcal{A}^{\sigma}_{\delta,r}}[\varrho : \underline{\mathbf{MP}}(\varrho) \geq \mathbf{sVal}(Q, \alpha_T) - \varepsilon] < 1$.*
- *For all strategies σ, there exist δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and a reward function r, such that $\mathbb{P}^{q_0}_{\mathcal{A}^{\sigma}_{\delta,r}}[\varrho : \underline{\mathbf{MP}}(\varrho) \geq \mathbf{sVal}(Q, \alpha_T)] < 1 - \gamma$.*

Proof sketch. Observe that the MEC is not a good EC. However, it does contain the GECs with states $\{q_1, q_2\}$ and $\{q_3, q_4\}$ respectively. Now, since those two GECs are separated by q_0, whose priority is 1, any winning strategy must at some point stop playing to q_0 and commit to a single GEC. Thus, the learning of the global EC can only last for a finite number of steps. It is then straightforward to argue that near-optimality with high-probability is the best achievable guarantee. ◀

4.3 General surely-good automata

In this section, we generalize our approach from single-EC automata to general automata. We will argue that, under a sure parity constraint, we can achieve a near-optimal mean payoff with high probability in any MEC (S, β) in which we end up with non-zero probability. That is, given that the event $\mathrm{Inf} \subseteq S$, defined as the set of all runs that eventually always stay within S, has non-zero probability measure.

▶ **Theorem 16.** *Consider a surely-good automaton $\mathcal{A} = (Q, A, T, p)$ and some $\pi_{\min} \in (0, 1]$. For all $\varepsilon, \gamma \in (0, 1)$ there exists a strategy σ such that for all $q_0 \in Q$, for all δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and all reward functions r, we have*

- *$\varrho \models \textsc{Parity}$ for all $\varrho \in \mathrm{Runs}^{q_0}(\mathcal{A}^{\sigma}_{\delta,r})$ and*
- *$\mathbb{P}^{q_0}_{\mathcal{A}^{\sigma}_{\delta,r}}[\varrho : \underline{\mathbf{MP}}(\varrho) \geq \mathbf{sVal}(S, \beta) - \varepsilon \mid \mathrm{Inf} \subseteq S] \geq 1 - \gamma$ for all $(S, \beta) \in \mathrm{MEC}_{\mathcal{A}_{\delta,r}}$ such that (S, β) is weakly good and $\mathbb{P}^{q_0}_{\mathcal{A}^{\sigma}_{\delta,r}}[\mathrm{Inf} \subseteq S] > 0$.*

Proof sketch. The strategy σ we construct follows a parity-winning strategy σ_{par} until a state contained in a weakly good MEC, that has not been visited before, is entered. In this case, the strategy follows τ (the strategy from Proposition 14). Observe that when τ switches to σ_{par} (a parity-winning strategy) it may exit the end component. If this happens, then the component is marked as visited and σ_{par} is followed until a new – not previously visited – maximal good end component is entered. In that case, we switch to τ once more. Crucially, the new strategy σ ignores MECs that are revisited ◀

▶ Remark (On the choice of MECs to reach). The strategy constructed for the proof of Theorem 16 has to deal with leaving a MEC due to the fallbacks to the parity-winning strategy σ_{par}. However, surprisingly, instead of actually following σ_{par}, upon entering a new MEC it has to restart the process of achieving a satisfactory mean-payoff. Indeed, otherwise the overall mass of sub-optimal runs from various MECs (each smaller than γ) could get concentrated in a single MEC, thus violating the advertised guarantees.

The strategy could be simplified as follows. First, we follow any strategy to reach a bottom MEC (BMEC) – that is, a MEC from which no other MEC is reachable. By definition, the winning strategy can be played here and the MEC cannot be escaped. Therefore, in the BMEC we run the strategy as described, and after the fallback we indeed simply follow σ_{par}. If we did not reach a BMEC after a long time, we could switch to the fallback, too. While this strategy is certainly simpler, our general strategy has the following advantage. Intuitively, we can force the strategy to stay in any current good MEC, even if it is not bottom, and thus maybe achieve a more satisfactory mean-payoff. Further, whenever we want, we can force the strategy to leave the current MEC and go to a lower one. For instance, if the current estimate of the mean payoff is lower than what we hope for, we can try our luck in a lower MEC. We further comment on the choice of unknown MECs in the conclusions.

5 Learning for MP under an Almost-Sure Parity Constraint

In this section, we turn our attention to learning strategies that must ensure a parity objective not with certainty (as in previous section) but *almost surely*, i.e. with probability 1. As winning almost surely is less stringent, we can hope both for a stricter yardstick (i.e. better target values) and also better ways of achieving such high values. We show here that this is indeed the case. Additionally, we argue that several important learning results can now be obtained with finite-memory strategies.

As previously, we make the hypothesis that we have removed from $\mathcal{A}$ all states from which the parity objective cannot be forced with probability 1 (no such state can ever be entered). Note that to compute the set of states to remove, we do not need the knowledge of δ but only the support as given by $\mathcal{A}$. States to remove can be computed in polynomial time using graph-based algorithms (see, e.g., [5]). An automaton $\mathcal{A}$ which contains only almost-surely winning states for the parity objective is called *almost-surely good*.

We have, as in the previous section, that for all automata $\mathcal{A}$ there exists a memoryless deterministic strategy σ such that for all $q_0 \in Q$, for all δ compatible with $\mathcal{A}$, for all r, the measure of the subset of $\varrho \in \mathrm{Runs}^{q_0}(\mathcal{A}^{\sigma}_{\mu,r})$ such that $\varrho \models \textsc{Parity}$ is equal to 1 (see e.g. [5]). Such a strategy is said to be *uniformly almost-sure winning (for the parity objective)*. In the sequel, we denote such a strategy $\sigma^{\mathrm{as}}_{\mathrm{par}}$.

We now study the design of learning strategies for mean-payoff optimization under *almost-sure* parity constraints for increasingly complex cases.

5.1 The case of a good end component

Consider an automaton $\mathcal{A} = (Q, A, T, p)$ such that (Q, α_T) is a GEC, and some $\pi_{\min} \in (0, 1]$.

Yardstick. For this case, we use as a yardstick the optimal expected mean-payoff value: $\mathbf{Val}(Q, \alpha_T) = \sup_\sigma \mathbb{E}^{q_0}_{\mathcal{A}^\sigma_{\delta,r}} [\underline{\mathbf{MP}}]$ for any $q_0 \in Q$.

Learning strategies. We start by noting that σ_∞ from Section 3 also ensures that the parity objective is satisfied almost surely when exercised in a GEC.

▶ **Proposition 17.** *One can compute a sequence $(L_i, O_i)_{i \in \mathbb{N}}$ such that for the resulting strategy σ_∞ we have that for all $q_0 \in Q$, for all δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and for all reward functions r, we have $\mathbb{P}^{q_0}_{\mathcal{A}^{\sigma_\infty}_{\delta,r}} [\textsc{Parity}] = 1$ and $\mathbb{P}^{q_0}_{\mathcal{A}^{\sigma_\infty}_{\delta,r}} [\varrho : \underline{\mathbf{MP}}(\varrho) \geq \mathbf{Val}(Q, \alpha_T)] = 1$.*

Proof. By Proposition 10, one can choose parameter sequences such that $L_i \geq |Q|$ for all $i \in \mathbb{N}$ and so that we obtain the second part of the claim. Then, since in every episode we have a non-zero probability of visiting a minimal even priority state, we obtain the first part of the claim as a simple consequence of the second Borel-Cantelli lemma. ◀

We now turn to learning using finite memory only. Consider parameters $\varepsilon, \gamma \in (0, 1)$. Let $\eta = \min\{\pi_{\min}, \eta_{\varepsilon/4}\}$ for $\eta_{\varepsilon/4}$ as defined for Lemma 5. The strategy τ_{fin} that we construct does the following.

1. First, it computes δ' such that $\delta' \sim^\eta \delta$ with probability at least $1 - \gamma$ and a reward function r' by following an exploration strategy λ for α_T during J steps (see Lemma 7).
2. Then, it computes a unichain deterministic expectation-optimal strategy $\sigma^{\delta'}_{\text{MP}}$ for $\mathcal{A}_{\delta',r'}$ and repeats the following forever: follow $\sigma^{\delta'}_{\text{MP}}$ for O steps, then follow λ for $|Q|$ steps.

Using the fact that, in a finite MC with a single BSCC, almost all runs obtain the expected mean payoff and the assumption that the EC is good, one can then prove the following result.

▶ **Proposition 18.** *For all $\varepsilon, \gamma \in (0, 1)$ one can compute $L, O \in \mathbb{N}$ such that for the resulting strategy τ_{fin}, for all $q_0 \in Q$, for all δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and for all reward functions r, we have $\mathbb{P}^{q_0}_{\mathcal{A}^{\tau_{\text{fin}}}_{\delta,r}} [\textsc{Parity}] = 1$ and $\mathbb{P}^{q_0}_{\mathcal{A}^{\tau_{\text{fin}}}_{\delta,r}} [\varrho : \underline{\mathbf{MP}}(\varrho) \geq \mathbf{Val}(Q, \alpha_T) - \varepsilon] \geq 1 - \gamma$.*

Optimality. Obviously, the result of Proposition 17 is optimal as we obtain the best possible value with probability one. We claim that the result of Proposition 18 is also optimal as we have seen that when we use finite learning, we cannot do better than ε-optimality with high probability, this can be proved on the example of Fig. 2 with a similar argument to the one that has been developed for the proof of Proposition 15.

5.2 The case of a single end component

Consider an almost-surely-good automaton $\mathcal{A} = (Q, A, T, p)$ such that (Q, α_T) is an EC and some $\pi_{\min} \in (0, 1]$. The EC is not necessarily good but as the automaton is almost-surely-good, then we have the analogue of Lemma 13 in this context.

▶ **Lemma 19.** *For all almost-surely-good automata $\mathcal{A} = (Q, A, T, p)$ such that (Q, α_T) is an EC there exists $(S, \beta) \subseteq (Q, \alpha_T)$ such that (S, β) is a GEC in $\mathcal{A}_{\delta,r}$ for all δ compatible with $\mathcal{A}$ and all reward functions r, i.e. (Q, α_T) is weakly good.*

Yardstick. As a yardstick for this case, we use the following value: $\mathbf{asVal}(Q, \alpha_T) \stackrel{\text{def}}{=} \max\{\mathbf{Val}(S, \beta) \mid (S, \beta) \subseteq (Q, \alpha_T) \text{ and } (S, \beta) \text{ is a GEC}\}$. That is, $\mathbf{asVal}(Q, \alpha_T)$ is the best expected mean-payoff value that can be obtained in a GEC included in the EC. Such a good EC exists by Lemma 19.

Learning strategy. We will now prove an analogue of Proposition 14. For any given $\varepsilon, \gamma \in (0, 1)$ we define the strategy σ as follows.

1. First, it follows an exploration strategy λ for α_T during sufficiently many steps (say K) to compute an approximation δ' of δ such that $\delta' \sim^{\eta_{\varepsilon/4}} \delta$ with probability at least $1 - \gamma/2$; and a reward function r' (see Lemma 7).
2. Next, it selects a GEC (S, β) with maximal value $\pm\frac{\varepsilon}{4}$ (see Lemma 19) and computes for it a strategy τ as in Proposition 18 with $\varepsilon_1/2$ and $\gamma/2$.
3. Finally, σ follows λ until (S, β) is reached, then switches to τ.

▶ **Proposition 20.** *For all $\varepsilon, \gamma \in (0, 1)$ one can construct a finite-memory strategy σ such that for all $q_0 \in Q$, for all δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and for all reward functions r, we have $\mathbb{P}^{q_0}_{\mathcal{A}^\sigma_{\delta,r}}[\textsc{Parity}] = 1$ and $\mathbb{P}^{q_0}_{\mathcal{A}^\sigma_{\delta,r}}[\varrho : \underline{\mathbf{MP}}(\varrho) \geq \mathbf{asVal}(Q, \alpha_T) - \varepsilon] \geq 1 - \gamma$.*
See the remark in Sect. 4.3 for a comment on the finite-memory implementability of σ.

Optimality. Using the same example and reasoning as in Proposition 15, we can show that this result is optimal and cannot be improved. Also note that using infinite memory would not help as shown with the example of Fig. 2, where the learning needs to be finite and enforcing the almost sure parity does not require infinite memory.

5.3 General almost-surely-good automata

We now generalize our approach to general almost-surely-good automata.

▶ **Theorem 21.** *Consider an almost-surely-good automaton $\mathcal{A} = (Q, A, T, p)$ and some $\pi_{\min} \in (0, 1]$. For all $\varepsilon, \gamma \in (0, 1)$ one can compute a finite-memory strategy σ such that for all $q_0 \in Q$, for all δ compatible with $\mathcal{A}$ and $\pi_{\min}$, and all reward functions r, we have*

- $\mathbb{P}^{q_0}_{\mathcal{A}^\sigma_{\delta,r}}[\textsc{Parity}] = 1$ *and*
- $\mathbb{P}^{q_0}_{\mathcal{A}^\sigma_{\delta,r}}[\varrho : \underline{\mathbf{MP}}(\varrho) \geq \mathbf{asVal}(S, \beta) - \varepsilon \mid \mathrm{Inf} \subseteq S] \geq 1 - \gamma$ *for all $(S, \beta) \in \mathrm{MEC}_{\mathcal{A}_{\delta,r}}$ such that (S, β) is weakly good and $\mathbb{P}^{q_0}_{\mathcal{A}^\sigma_{\delta,r}}[\mathrm{Inf} \subseteq S] > 0$.*

Proof sketch. The argument to prove the above result is simple: σ follows a strategy $\sigma^{\text{as}}_{\text{par}}$ that ensures satisfying the parity objective almost surely. Then, if the run reaches a state contained in a weakly good MEC, σ switches to τ as described in Proposition 20. ◀

See the remark in Sect. 3.1 for a word on how to modify σ to favour some unknown MECs.

6 Conclusion

As future work, we would like to study different configurations resulting from relaxations of the assumptions we make in this work (i.e. full support, $\pi_{\min}$, and bounded reward). Further, we would like to obtain model-free learning algorithms ensuring the same guarantees we give here. Finally, we have left open the choice of strategy driving the visits to MECs in Theorems 16 and 21 (as long as it satisfies the parity objective). Indeed, the question of computing an "optimal" such strategy in view of the unknown components of the MDP can be addressed in different ways. One such way would be to model the problem as a Canadian traveler problem [24].

References

1. Shaull Almagor, Orna Kupferman, and Yaron Velner. Minimizing expected cost under hard boolean constraints, with applications to quantitative synthesis. In Josée Desharnais and Radha Jagadeesan, editors, *27th International Conference on Concurrency Theory, CONCUR 2016, August 23-26, 2016, Québec City, Canada*, volume 59 of *LIPIcs*, pages 9:1–9:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.CONCUR.2016.9.
2. Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. *CoRR*, abs/1708.08611, 2017. arXiv:1708.08611.
3. Benjamin Aminof and Sasha Rubin. First-cycle games. *Information and Compution*, 254:195–216, 2017. doi:10.1016/j.ic.2016.10.008.
4. Krzysztof R. Apt and Erich Grädel. *Lectures in game theory for computer scientists*. Cambridge University Press, 2011.
5. Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.
6. Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 449–458. PMLR, 2017. URL: http://proceedings.mlr.press/v70/bellemare17a.html.
7. Julien Bernet, David Janin, and Igor Walukiewicz. Permissive strategies: from parity games to safety games. *ITA*, 36(3):261–275, 2002. doi:10.1051/ita:2002013.
8. Raphaël Berthon, Mickael Randour, and Jean-François Raskin. Threshold constraints with guarantees for parity objectives in Markov decision processes. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 121:1–121:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.121.
9. Tomás Brázdil, Krishnendu Chatterjee, Martin Chmelik, Vojtech Forejt, Jan Kretínský, Marta Z. Kwiatkowska, David Parker, and Mateusz Ujma. Verification of markov decision processes using learning algorithms. In *Automated Technology for Verification and Analysis - 12th International Symposium, ATVA 2014, Sydney, NSW, Australia, November 3-7, 2014, Proceedings*, volume 8837 of *Lecture Notes in Computer Science*, pages 98–114. Springer, 2014.
10. Tomás Brázdil, Antonín Kucera, and Petr Novotný. Optimizing the expected mean payoff in energy Markov decision processes. In *Automated Technology for Verification and Analysis - 14th International Symposium, ATVA 2016, Chiba, Japan, October 17-20, 2016, Proceedings*, volume 9938 of *Lecture Notes in Computer Science*, pages 32–49, 2016. doi:10.1007/978-3-319-46520-3.
11. Véronique Bruyère, Emmanuel Filiot, Mickael Randour, and Jean-François Raskin. Meet your expectations with guarantees: Beyond worst-case synthesis in quantitative games. In Ernst W. Mayr and Natacha Portier, editors, *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014), STACS 2014, March 5-8, 2014, Lyon, France*, volume 25 of *LIPIcs*, pages 199–213. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014. doi:10.4230/LIPIcs.STACS.2014.199.
12. Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasipolynomial time. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 252–263. ACM, 2017. doi:10.1145/3055399.3055409.

13 Krishnendu Chatterjee. Concurrent games with tail objectives. *Theoretical Computer Science*, 388(1-3):181–198, 2007. `doi:10.1016/j.tcs.2007.07.047`.

14 Krishnendu Chatterjee. Robustness of structurally equivalent concurrent parity games. In Lars Birkedal, editor, *Foundations of Software Science and Computational Structures - 15th International Conference, FOSSACS 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 - April 1, 2012. Proceedings*, volume 7213 of *Lecture Notes in Computer Science*, pages 270–285. Springer, 2012. `doi:10.1007/978-3-642-28729-9_18`.

15 Krishnendu Chatterjee and Laurent Doyen. Energy and mean-payoff parity markov decision processes. In *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings*, volume 6907 of *Lecture Notes in Computer Science*, pages 206–218. Springer, 2011. `doi:10.1007/978-3-642-22993-0`.

16 Krishnendu Chatterjee, Petr Novotný, Guillermo A. Pérez, Jean-François Raskin, and Dorde Zikelic. Optimizing expectation with guarantees in POMDPs. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3725–3732. AAAI Press, 2017. URL: `http://www.aaai.org/Library/AAAI/aaai17contents.php`.

17 Lorenzo Clemente and Jean-François Raskin. Multidimensional beyond worst-case and almost-sure problems for mean-payoff objectives. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 257–268. IEEE Computer Society, 2015. `doi:10.1109/LICS.2015.33`.

18 Przemyslaw Daca, Thomas A. Henzinger, Jan Kretínský, and Tatjana Petrov. Faster statistical model checking for unbounded temporal properties. In *Tools and Algorithms for the Construction and Analysis of Systems - 22nd International Conference, TACAS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, volume 9636 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2016. `doi:10.1007/978-3-662-49674-9`.

19 Alexandre David, Peter Gjøl Jensen, Kim Guldstrand Larsen, Axel Legay, Didier Lime, Mathias Grund Sørensen, and Jakob Haahr Taankvist. On time with minimal expected cost! In *Automated Technology for Verification and Analysis - 12th International Symposium, ATVA 2014, Sydney, NSW, Australia, November 3-7, 2014, Proceedings*, volume 8837 of *Lecture Notes in Computer Science*, pages 129–145. Springer, 2014.

20 Hugo Gimbert. Pure stationary optimal strategies in Markov decision processes. In Wolfgang Thomas and Pascal Weil, editors, *STACS 2007, 24th Annual Symposium on Theoretical Aspects of Computer Science, Aachen, Germany, February 22-24, 2007, Proceedings*, volume 4393 of *Lecture Notes in Computer Science*, pages 200–211. Springer, 2007. `doi:10.1007/978-3-540-70918-3_18`.

21 Sebastian Junges, Nils Jansen, Christian Dehnert, Ufuk Topcu, and Joost-Pieter Katoen. Safety-constrained reinforcement learning for MDPs. In Marsha Chechik and Jean-François Raskin, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 22nd International Conference, TACAS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, volume 9636 of *Lecture Notes in Computer Science*, pages 130–146. Springer, 2016. `doi:10.1007/978-3-662-49674-9_8`.

22 Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *J. Artif. Intell. Res.*, 4:237–285, 1996. `doi:10.1613/jair.301`.

23 James R. Norris. *Markov chains*. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, 1998.

24 Christos H. Papadimitriou and Mihalis Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 84(1):127–150, 1991.

25 Martin L. Puterman. *Markov Decision Processes*. Wiley-Interscience, 2005.

26 Stuart J. Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach (3. internat. ed.)*. Pearson Education, 2010. URL: `http://vig.pearsoned.com/store/product/1,1207,store-12521_isbn-0136042597,00.html`.

27 Eilon Solan. Continuity of the value of competitive Markov decision processes. *Journal of Theoretical Probability*, 16(4):831–845, 2003.

28 Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning. MIT Press, 2018. URL: `http://www.incompleteideas.net/book/the-book-2nd.html`.

29 Wolfgang Thomas. On the synthesis of strategies in infinite games. In *STACS*, pages 1–13, 1995. `doi:10.1007/3-540-59042-0_57`.

30 Leslie G. Valiant. A theory of the learnable. In Richard A. DeMillo, editor, *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, pages 436–445. ACM, 1984. `doi:10.1145/800057.808710`.

31 Moshe Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 327–338. IEEE Computer Society, 1985. `doi:10.1109/SFCS.1985.12`.

32 Christopher J. C. H. Watkins and Peter Dayan. Technical note q-learning. *Machine Learning*, 8:279–292, 1992. `doi:10.1007/BF00992698`.

33 Min Wen, Rüdiger Ehlers, and Ufuk Topcu. Correct-by-synthesis reinforcement learning with temporal logic constraints. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015, Hamburg, Germany, September 28 - October 2, 2015*, pages 4983–4990. IEEE, 2015. URL: `http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7347169`.

34 Min Wen and Ufuk Topcu. Probably approximately correct learning in stochastic games with temporal logic specifications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3630–3636. IJCAI/AAAI Press, 2016. URL: `http://www.ijcai.org/Proceedings/2016`.