Computing Optimal Strategies for Markov Decision Processes with Parity and Positive-Average Conditions

Hugo Gimbert[1], Youssouf Oualhadj[2], and Soumya Paul[3]

[1] LaBRI, CNRS, Université Bordeaux 1, France
hugo.gimbert@labri.fr
[2] LaBRI, Université Bordeaux 1, France
youssouf.oualhadj@labri.fr
[3] The Institute of Mathematical Sciences, Chennai, India
soumya@imsc.res.in

Abstract. We study Markov decision processes (one-player stochastic games) equipped with parity and positive-average conditions. In these games, the goal of the player is to maximize the probability that both the parity and the positive-average conditions are fulfilled. We show that the values of these games are computable in polynomial time. We also show that optimal strategies exist, require only finite memory and can be effectively computed.

1 Introduction

Infinite turn-based games are useful tools for modeling and analyzing open reactive systems. Such games have been extensively studied, both in the deterministic and the stochastic setting. In the deterministic setting, there are usually 2 players, player 0 or the system player and player 1 or the environment player who take turns in making moves in an arena, thus producing an infinite sequence of moves called a play. The objective is given in terms of a Borel subset Φ of the set of plays. The aim of player 0 is to play in such a manner that the resulting play always lies in Φ. The aim of player 1 is the opposite: to foil player 0's aim and to make the play settle down outside Φ.

A correct controller of the system ensures the system behaves correctly in a demonic environment. Synthesizing such a controller amounts to computing a winning strategy σ for player 0: no matter how player 1 plays, player 0 always wins if she plays according to σ.

Markov decision processes. In systems where hardware failures and other random events occur, the behavior of the environment is typically represented as a stochastic process [17, 21]. Markov decision processes (MDP's) have proven to be a powerful [16, 1] yet algorithmically tractable [7] tool. In Markov decision processes, the environment no longer is a strategic player but its moves are chosen randomly according to fixed transition probabilities that depend on the current state of the system. This setting naturally extends to 2 $\frac{1}{2}$ player games but in this paper we focus on MDP's i.e. 1 $\frac{1}{2}$ player games.

Winning almost-surely. In Markov decision processes there are generally few chances that player 0 has a strategy to win *for sure*, player 0 rather tries to maximize the probability to fulfill her winning objective. The best strategies for player 0 are *almost-surely* winning strategies, which guarantee a win with probability 1. The existence of almost-surely winning strategies is the solution concept we are focusing on in this paper.

The parity and positive average objectives. The class of ω-regular objectives is a sub-class of Borel objectives that naturally arise from logical specifications of reactive systems. The *parity* objective is an objective that is complete for ω-regular objectives, in the sense that it can express any such objective [15]. In a parity game, with every vertex is associated an integer called a *priority*. Player 0 has to play in such a way that the maximum priority visited infinitely often is even. The parity objective enjoys the luxury of being positionally determined, that is, either of the players always has a winning strategy that is independent of the history of the play (see e.g. [22, 6, 23]). The interest for parity games also stems from the close relation between parity games and μ-calculus [9].

When dealing with performance evaluation, one needs to go beyond ω-regular objectives. This is needed in particular to measure the average performances of the system along a play. In MDPs equipped with mean-payoff or positive average objectives, every move is associated with a real number called a reward and Player 0 seeks to maximize the average reward along the play.

The *Positive Average* objective states that Player 0 is declared the winner if the average value of the flow of rewards is strictly positive. This slightly differs from the *mean-payoff objective* where player 0 rather tries to maximize the expectation of the average value of the rewards.

The definition of positive average and mean-payoff objectives seem very close to each other, yet they are quite different when turning to applications. The positive average condition can be more suited to express certain constraints on the Quality of Service (QoS) of critical systems. For example, it may be more desirable that a car speed controller reacts to any emergency event in 100 ms almost-surely rather than in 1 ms with probability 99% and 9000 ms with probability 1%. This preference is easily expressible using the positive average objective (reacting in less than 150 ms is fulfilled with probability 100% in the first case and only 99% in the second case) whereas the mean-payoff objective is maximized in the second case.

To guarantee both correct behavior and maximal performances it is necessary to consider boolean combinations of parity and positive-average conditions.

Our results. In this work we consider Markov decision processes equipped with the conjunction of parity and positive average conditions. Our main result is that in such games, the values are computable in polynomial time. Moreover, there exist optimal strategies with finite memory and these strategies are effectively computable. Finally, our definition of positive average MDPs is robust, since these results hold both under the liminf and the limsup semantics.

Comparison with previous work. Apart from parity positive average games, there has been other attemps to define winning conditions that mix both parity conditions and conditions about the mean value of rewards.

In mean-payoff parity games [5] the objective of the player is to maximize her long run average reward and satisfy the parity condition: in case the parity condition holds, the player's income is the mean value of rewards, in the opposite case it is $-\infty$. This objective is hard to rephrase in the stochastic setting because to be able to compute expectations, one would have to arbitrarily choose some punishing constants to replace the $-\infty$ payoff with. Moreover, as is shown by the car speed controller example, the positive average and the mean-payoff conditions may induce different preference orders on the set of outcomes. For these two reasons, it is hard to compare our results with the results of [5].

In an Energy Parity game [4], the player not only should win the parity game but she should ensure at the same time that her cumulative reward never goes below 0. It seems that in the deterministic setting the energy and the positive average conditions are closely related: if the player wins the positive average game, then there exists an initial amount of energy such that the player wins the energy game as well. However this is no more true in the stochastic setting, where we do not know any precise link between energy parity games and parity and positive average games.

Priority mean-payoff games [13] are another attempt to mix parity and mean-payoff games. Here, the payoff associated to a play is the average value of the stream of payoffs seen at those moments where the highest priority seen infinitely often was visited. These games have some nice properties: there always exists *positional* optimal strategies in priority mean-payoff games and moreover these games are closely linked to discounted games [13, 14]. However, priority mean-payoff games seem quite incomparable with parity and positive average games, both algorithmically and semantically.

Organisation of the paper. In section 2, we introduce basic notions and some results about MDPs together with some material specific to the reachability and tail winning conditions. In section 3, we introduce parity and positive average games. In section 4, we present our main result (Theorem 4 and Corollary 3) about the polynomial time computability of values of parity and positive average games. Moreover we prove that finite memory strategies are sufficient (Theorem 5). In section 5, we prove that using the $\liminf$ semantic does not change the values of parity and positive average Markov decisions processes.

2 Preliminaries

In this section we introduce several basic notions that will be used throughout the paper: game arenas, strategies, games and values and, finally, optimal, positively and almost-surely winning strategies.

2.1 Games and strategies

An arena describes the set of moves that players are allowed to use when playing the game.

Definition 1 (Markov decision processes and plays). *A Markov decision process or arena is a tuple $\mathcal{M} = (V, V_0, V_R, E, \delta)$ where: (V, E) is a directed graph without deadlocks, the set of vertices V is partioned into two parts,* Player 0 *vertices V_0 and chance vertices V_R. We equip $\mathcal{M}$ with the function $\delta : V_R \to \mathcal{D}(V)$ that assigns to each chance vertex a probability distribution over V, such that $\delta(u)(v) > 0$ if and only if $(u, v) \in E$. We denote by vE the set $\{u \in V \mid (v, u) \in E\}$ of successors of a vertex v.*

A play in the arena $\mathcal{M}$ is a sequence of vertices $v_0 v_1 \cdots v_n$ such that for every $0 \leq i < n$, $(v_i, v_{i+1}) \in E$.

Definition 2 (Subarenas). *A subgraph (V', E') of (V, E) is a subarena if and only if each vertex in V'_0 has a successor in V', and all the successors of a vertex in V'_R are in V'. We usually denote $\mathcal{A}[V']$ the subarena obtained by the set V'.*

Definition 3 (Games and winning conditions). *Let V be a set of vertices. A winning condition is a Borel subset $\Phi \subseteq V^\omega$. An infinite play is winning for* Player 0 *if it belongs to Φ. A game is a tuple $(\mathcal{M}, \Phi)$ made of an arena $\mathcal{M}$ with vertices V and a winning condition Φ.*

A strategy of the player tells her how to play the game. Formally,

Definition 4 (Strategy). *A strategy σ for* Player 0 *is a function $\sigma : V^* V_0 \to V$ such that, for every finite play $w = v_0 v_1 \ldots v_k \in V^* V_0$, $\sigma(w) \in v_k E$.*

Strategies with finite memory are finitely representable strategies.

Definition 5 (Finite memory and positional strategies). *A strategy σ is finite memory if there exists a finite set M, the memory of the strategy, an initial memory $m_I \in M$, and functions $g : M \times V \to M$ and $h : M \times V \to V$ such that if $v_0 \ldots v_k \in V^* V_0$ is a history and $m_0 \ldots m_{k+1}$ is a sequence determined by $m_0 = m_I$ and $m_{i+i} = g(m_i, v_i)$ then $\sigma(v_0 \ldots v_k) = h(m_{k+1}, v_k)$.*

A strategy σ is said to be positional if M is a singleton. A positional strategy σ can be specified just by a function from the set of vertices of Player 0 *to the set of all vertices, that is, $\sigma : V_0 \to V$ such that for all $v \in V_0$, $\sigma(v) \in vE$.*

In a Markov decision process $\mathcal{M}$, once we have fixed a strategy σ for Player 0 and an initial vertex v, this defines naturally a probability measure $\mathbb{P}_v^\sigma$ on V^ω.

2.2 Values, optimal strategies and almost-surely winning strategies

For a winning condition $\Phi \subseteq V^\omega$, the value of a vertex $v \in V$ with respect to strategy σ is denoted:

$$\mathrm{val}_\sigma(q) = \mathbb{P}_v^\sigma(\Phi) \ ,$$

intuitively this is the probability that Player 0 wins if the play is consistent with the strategy σ.

Definition 6 (Values and optimal strategies). *The value of a vertex is defined as*

$$\mathrm{val}(v) = \sup_{\sigma} \mathrm{val}_{\sigma}(v)$$

If a strategy σ for player 0 is such that $\mathrm{val}(v) = \mathrm{val}_{\sigma}(v)$ then σ is said to be optimal.

There is an alternative notion of solution to a stochastic game, which is more qualitative.

Definition 7 (Almost-surely and positively winning strategies). *We say that* Player 0 *wins almost-surely (resp. positively) from a vertex v if she has a strategy σ such that $\mathbb{P}_v^{\sigma}(\Phi) = 1$ (resp. $\mathbb{P}_v^{\sigma}(\Phi) > 0$). In this case, vertex v is said to be almost-surely (resp. positively) winning for* Player 0*. The set of almost-surely (resp. positively) winning vertices for* Player 0 *is denoted $W_{=1}$ (resp. $W_{>0}$) and called the almost-surely (resp. positively) winning region of* Player 0*.*

A positively winning vertex is also positively losing.

Definition 8. *A Markov decision process is qualitatively determined if every vertex is either almost-surely winning or positively losing for* Player 0*.*

2.3 Reachability games

The simplest class of games is the class of reachability games. In a reachability game, the goal of the player is to reach a set of target states $T \subseteq V$, in other words the winning condition is V^*TV^ω.

In reachability games, the set of positively and almost-surely winning vertices is especially easy to compute, using elementary fixpoint algorithms.

Definition 9 (Positive and almost-sure attractors). *The positive attractor for player 0 to a subset T of V is the set of vertices from which the player can reach T with positive probability. It is denoted by $\mathrm{Attr}_{>0}(T, V)$ and is formally defined as follows. Let $f : 2^V \to 2^V$ be the operator such that for any $U \subseteq V$,*

$$f(U) = \{v \in V \mid vE \cap U \neq \emptyset\} \ .$$

Then $\mathrm{Attr}_{>0}(T, V)$ is the least fixed point of $f(T)$.

The positive attractor of the chance player to a subset T of V is denoted by $\mathrm{Attr}^c_{>0}(T, V)$ and is defined similarly.

The almost-sure attractor for player 0 to T is the set of vertices from which she can reach T with probability 1. It is denoted by $\mathrm{Attr}_{=1}(T, V)$ and is defined as

$$\mathrm{Attr}_{=1}(T, V) = (V \setminus \mathrm{Attr}^c_{>0}(V \setminus Y, V)) \cup T \ ,$$

where $Y = \mathrm{Attr}_{>0}(T, V)$.

Note that there is a positional strategy for player 0 to “attract” the play from any vertex in $\mathrm{Attr}_{>0}(T,V)$ (resp. $\mathrm{Attr}_{=1}(T,V)$) to T with positive probability (resp. probability 1). Properties of positive and almost-sure attractors are given in the following proposition.

Proposition 1. *The positive (resp. almost-sure) attractor of* Player 0 *to* T *is exactly the set of positively (resp. almost-surely) winning vertices of* Player 0 *in the reachability game.*

The complement of a positive attractor for Player 0 *is a trap for her.*

Definition 10 (Safe set). *Similar to the attractor set, we define the safe set of the player as the largest subarena from which the player has a strategy to avoid reaching* T*. Formally, it is denoted* $\mathrm{Safe}(T,V)$ *and obtained as follow. Let* $f : 2^V \to 2^V$ *such that for any subset* U *of* V

$$f(U) = \{v \in V_p \mid vE \subseteq U\} \cup \{v \in V_c \mid vE \cap U \neq \emptyset\} \cup T \enspace .$$

Then $\mathrm{Safe}(T,V)$ *is the complement of the least fixed point of* $f(T)$*.*

2.4 Tail Games

We say that a game is tail if the winning condition that equip this game is tail.

Definition 11 (Tail winning condition). *Let* $\Phi \in V^\omega$ *a winning condition.* Φ *is tail if* $\forall u \in V^*$ *and* $\forall w \in V^\omega$ *then*

$$uw \in \Phi \iff w \in \Phi \enspace .$$

For Markov decision processes equipped with tail winning condition, the notions of values and qualitative solutions are tightly linked. This is ilustrated by Theorem 1 which will be used throughout the paper.

Theorem 1. *[12] In any Markov decision process equipped with a tail winning condition, the three following assertions hold.*

1. *Tail games are qualitatively determined. winning vertices.*
2. *One can implement an optimal strategy with the same memory as needed by an almost-surely winning strategy.*
3. *The vertices with value 1 are exactly the almost-surely*

The above theorem shows that one can focus only on the computation of almost-surely winning regions when considering tail winning conditions on Markov decision processes. Proposition 2 is a general result that shows how to solve any disjunction of tail winning conditions.

Proposition 2. *Let* $\Phi_0, \cdots, \Phi_n$ *be* n *tail winning conditions and* $\mathcal{M}$ *a Markov decision process. The almost-surely winning region for the game* $\bigcup_{i=0}^{n} \Phi_i$ *is given by the set* $\mathrm{Attr}_{=1}(\bigcup_{i=0}^{n} W_{=1}[\Phi_i], V)$*.*

Proof. Let us prove that Player 0 has an almost-surely winning strategy from the set $\mathrm{Attr}_{=1}(\bigcup_{i=0}^{n} W_{=1}[\Phi_i], V)$. Player 0 plays as follows. First she applies her attractor strategy until she reaches one of the $W_{=1}[\Phi_i]$ then she applies her almost-surely winning strategy σ_i associated with the game Φ_i. This strategy is clearly almost-surely winning since Player 0 reaches one of the $W_{=1}[\Phi_i]$ with probability 1.

To see that Player 0 cannot win almost-surely outside $\mathrm{Attr}_{=1}(\bigcup_{i=0}^{n} W_{=1}[\Phi_i], V)$, let $W = \mathrm{Attr}_{=1}(\bigcup_{i=0}^{n} W_{=1}[\Phi_i], V)$ and consider the largest sub-arena $\mathcal{A}$ in $V \setminus W$. There, Player 0 has no strategy to win almost-surely for any Φ_i, by qualitative determinacy (see Theorem 1, fact 3) we get that $\forall v \in \mathcal{A},\ \forall \sigma,\ \forall 0 \leq i \leq n,\ \mathbb{P}_v^{\sigma}(\Phi_i) = 0$. This implies $\forall v \in \mathcal{A},\ \forall \sigma,\ \mathbb{P}_v^{\sigma}(\bigcup_{i=0}^{n} \Phi_i) = 0$, what shows that every vertex in $\mathcal{A}$ has value 0. For any other vertex v not in W and not in $\mathcal{A}$, the probability for a given strategy σ that the play reaches $\mathcal{A}$ is strictly greater than 0 otherwise it would imply that $v \in W$, which terminates the proof. □

Corollary 1. *Let Φ a tail winning condition. Assume that $W_{=1}[\Phi]$ can be computed in polynomial time, then there exists a polynomial time algorithm to compute the value of each vertex.*

Proof. The first fact of Theorem 1 implies that if Player 0 cannot win almost-surely then she cannot win positively, hence to win, Player 0 has to reach her almost-surely winning region. Using the third fact of the same theorem we get that in tail games the value of vertex is exactly the value of the reachability game played on the same arena and where the target set is the almost-surely winning region. Assuming that the almost-surely winning region can be computed in polynomial time and using the fact the reachability games can be solved in polynomial time finish the proof. □

3 Parity Games and Positive-Average Games

In this section, we define parity games, positive-average games, and recall known results about these games.

3.1 Parity games

In parity games, the winner of the play is determined by the set of priorities visited infinitely often during the play.

Definition 12 (Parity games). *Let C be a finite subset of $\mathbb{N}$ called the set of priorities. A parity game is played in a Markov decision process whose set of vertices V is labelled with a priority function $\chi : V \to C$ that assigns to each vertex a priority. The parity winning condition is:*

$$\mathrm{Par} = \{v_0 v_1 v_2 \cdots \in V^{\omega} \mid \limsup_n \chi(v_n) \textit{ is even}\} \ .$$

For any priority $d \in C$, we denote the set of vertices having priority d by V_d, that is, $V_d = \{v \in V \mid \chi(v) = d\}$. Parity games are a very useful tool in verification and automata theory [15].

Theorem 2 ([23, 6, 8]). *In Markov decision processes equipped with parity condition,* Player 0 *has a positional optimal strategy. Moreover, parity games are qualitatively determined and the almost-surely (resp. positively) winning regions are exactly the set of vertices with value* 1 *(resp. with strictly positive value). These values and the positional optimal strategies are computable in* polynomial *time.*

3.2 Positive average games

In positive average games, Player 0 wants to maximize the probability that the average value of rewards is strictly positive.

Definition 13 (Positive average games). *A positive average game is played in a Markov decision process whose set of vertices V is labelled with a reward mapping $r : V \to \mathbb{Q}$ that assigns to each vertex a rational number called the reward. The positive average winning condition is:*

$$\mathrm{Avg}_{>0} = \left\{ v_0 v_1 v_2 \cdots \in V^\omega \mid \limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} r(v_i) > 0 \right\} \quad . \tag{1}$$

There is another natural definition of positive average games, which is very similar except the lim sup is replaced by lim inf. We denote this condition $\underline{\mathrm{Avg}}_{>0}$.

We shall show later (cf. Theorem 6) that the choice of either definition does not impact our results for one-player stochastic games.

Theorem 3. *In a Markov decision process equipped with positive-average condition,* Player 0 *has a stationary optimal strategy. Moreover, parity games are qualitatively determined and the almost-surely (resp. positively) winning regions are exactly the set of vertices with value* 1 *(resp. with strictly positive value). These values and the positional optimal strategies are computable in* polynomial *time.*

To prove this theorem we use mean-payoff games and use the fact that these games can be solved in polynomial time using linear programming.

Definition 14 (Mean-payoff Games). *A mean-payoff game is played in a Markov decision process whose set of vertices V is labelled with a reward mapping $r : V \to \mathbb{Q}$ that assigns to each vertex a rational number called the reward. The value of a vertex v in a mean-payoff game is*

$$\mathrm{val}(v) = \sup_\sigma \mathbb{E}_v^\sigma \left[\limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} r(v_i) \right] \quad .$$

Proof (Theorem 3). Since positive-average games are tail, the first part of the theorem follows directly from Theorem 1.

Since the set of almost-surely winning vertices is exactly the set of vertices of value 1, computing the values amounts to compute the almost-surely winning region. In order to compute this set we first use linear programming to compute the set of vertices W with value > 0 for the mean-payoff game. This can be done in polynomial time [19]. Let τ an optimal strategy for the mean-payoff game, by [18, 10] we know that this strategy can be chosen memoryless, let also $\mathcal{M}[\tau]$ the Markov chain induced by τ. In $\mathcal{M}[\tau]$ consider C the set of all recurrent states. We show that Player 0 wins the positive-average game almost-surely from any vertex in $W = \mathrm{Attr}_{=1}(C, V)$. Let π the attraction strategy and let σ the following strategy, $\sigma(v) = \pi(v)$ if $v \in W \setminus C$ and $\sigma(v) = \tau(v)$ if $v \in C$. σ is clearly positional. Let us show that σ almost-surely winning, from every vertex $v \in W$ Player 0 reaches C with probability 1. Second from for each vertex $c \in C$ we can write $\mathbb{E}_c^\tau \left[\limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} r(c_i)\right] > 0$, since c is recurrent using the strong law of large numbers we get that $\mathbb{P}_c^\tau \left(\limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} r(c_i) > 0\right) = 1$, which shows that σ is almost-surely winning.

Let us show that Player 0 cannot win almost-surely outside W. We know that from any vertex outside W, the probability to reach a recurrent state c such that $\mathbb{E}_c^\tau \left[\limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} r(c_i)\right] \leq 0$ is strictly greater than 0, hence the value of every vertex outside W is stricly less than 1. Corollary 1 concludes the proof. □

Corollary 2. *In any Markov decision process $\mathcal{M}$ we have:*

$$\forall v \in V, \;\; \mathrm{val}_{\mathrm{Avg}_{>0}}(v) = \mathrm{val}_{\underline{\mathrm{Avg}}_{>0}}(v) \;\; .$$

The proof of this corollary is postponed to Section 5 where it is proved for a larger class of games.

4 Solving Parity and Positive-average Games with lim sup semantics

4.1 Computing the Values

In this section we consider Markov decision processes equipped with $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ winning condition. We give a polynomial time algorithm that computes the almost-surely winning region and we show that Player 0 needs memory of size exponential in the size of the arena.

We characterize the winning regions by induction on the priorities available in the arena. The two following lemmata proved in the appendix characterize the almost-surely winning regions when the highest priority is even (Lemma 1) and when the highest priority is odd (Lemma 2).

Lemma 1. *In any Markov decision process $\mathcal{M}$ where the winning condition is* $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ *and the highest priority d is even, the set of almost-surely winning vertices is given by the largest set $W \subseteq V$ such that*

1. *$\mathcal{A}[W]$ is a subarena of $\mathcal{M}$.*
2. Player 0 *wins almost surely the* $\mathrm{Avg}_{>0}$ *played in the subarena $\mathcal{A}[W]$,*
3. Player 0 *wins almost surely the* $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ *game played in the subarena* $\mathcal{A}[W \setminus \mathrm{Attr}_{>0}(V_d \cap W, W)]$.

The core idea of Lemma 1 is that in the set $\mathcal{A}[W \setminus \mathrm{Attr}_{>0}(V_d \cap W, W)]$ Player 0 has an almost-surely winning strategy to achieve the $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ objective, hence by being consistent with it she wins. In the set $\mathrm{Attr}_{>0}(V_d \cap W, W)]$, Player 0 applies in turn the attraction strategy to ensure the parity objective and her almost-surely winning strategy for $\mathrm{Avg}_{>0}$ objective until her accumulated reward gets strictly greater than some integer. By doing so Player 0 is ensured that after one alternation of the attraction strategy and the $\mathrm{Avg}_{>0}$ strategy, the sum of her average reward is always strictly positive.

Lemma 2. *In any Markov decision process $\mathcal{M}$ where the winning condition is* $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ *and the highest priority d is odd, the set of almost-surely winning vertices is given by* $\mathrm{Attr}_{=1}(R, V)$*, where R is the almost-surely winning region for the* $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ *game played in the subarena* $\mathcal{A}[\mathrm{Safe}(V_d, V)]$.

Theorem 4. *In any Markov decision process $\mathcal{M}$ where the winning condition is* $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$*, the almost-surely winning region is computable in* polynomial *time.*

Proof. To compute the values in polynomial time we use similar technics as in the proof of Theorem 2. For each even priority d, we create a new game Φ_d played on the same arena where each priority d is transformed into 2, each priority greater than d is transformed into 3 and all the priorities less than d are transformed into 1. Then to solve each of these games we use the procedure described in Algorithm 1.

Let us show that Algorithm 1 is correct. First it considers the largest subarena where the highest priority is 2, namely $\mathcal{A}[S]$. In $\mathcal{A}[S]$, once a fixed point is reached, R' satisfies the conditions of Lemma 1. Finally it uses Lemma 2 to compute the almost-surely winning region in the whole arena. Using the fact that the original game can be rewritten as the disjunction of all the Φ_d Proposition 2 shows that the almost-surely winning region is given by $\mathrm{Attr}_{=1}(\bigcup_{d \in D} W_{=1}[\Phi_d])$, where D is the set of even priorities.

We now argue on the running time complexity. Each Φ_d can be solved in polynomial time, hence our procedure runs in $\mathcal{O}(|D| \cdot L)$ where L is the time one needs to solve each Φ_d. □

From the above theorem and by Corollary 1, we get the following corollary.

Corollary 3. *In any Markov decision process $\mathcal{M}$ where the winning condition is* $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$*, the values are computable in* polynomial *time.*

Algorithm 1 Computes the almost-surely winning region for Player 0 in one of the game Φ_d

```
$R \leftarrow \emptyset$
$S \leftarrow \mathrm{Safe}(V_3, V)$
repeat
    In the arena $\mathcal{A}[S]$ compute $R'$ the almost-surely winning region for Player 0 for $\mathrm{Avg}_{>0}$ game
    In the arena $\mathcal{A}[R']$ compute $R''$ the positively losing region for the parity game for Player 0.
    $S \leftarrow Safe(R'', S)$
until $R'' = \emptyset$
return $\mathrm{Attr}_{=1}(R', V)$
```

4.2 Finite Memory strategies are sufficient

In this subsection, we examine the memory needed by Player 0 to win almost-surely. We prove the following theorem.

Theorem 5. *For* Player 0*, memory of size exponential in the size of the arena is sufficient and necessary to implement her almost-surely winning strategies.*

To prove this theorem, we first prove Proposition 3 where it is shown that almost-surely winning strategies with finite memory exist, in Proposition 4 we show that exponential size memory is sufficient and in the example of Figure 1 it is shown that exponential size memory is necessary.

The main idea behind the proof of Theorem 5 is that instead of applying her positive-average strategy until her reward goes above some well chosen integer, Player 0 will apply this strategy until she is ensured that her expected average reward becomes greater than some value.

Existence of Finite Memory Strategies To establish the existence of finite memory strategies, we need the following two lemmata that show the relationship between the total-reward defined as follow $\liminf_{n\to\infty} \sum_{i=0}^{n} r(v_i)$ and the average reward (see Definition 13).

Lemma 3 ([2]). *Let $\mathcal{M}$ be a finite irreducible Markov chain with reward. Let s be a recurrent state of $\mathcal{M}$. Assume $\liminf_{n\to\infty} \sum_{i=0}^{n} r(v_i) = \infty$. Then there exists an $\eta > 0$ such that the average reward accumulated between two consecutive visits of s is at least η. Moreover η has polynomial bit complexity and depends on the number of states in $\mathcal{M}$.*

Lemma 4 ([2]). *In any Markov decision process $\mathcal{M}$,* Player 0 *has a positional strategy σ such that*

$$\forall v \in W,\ \mathbb{P}_v^{\sigma}\left(\liminf_{n} \sum_{i=0}^{n} r(v_i) = \infty\right) = 1 \ , \tag{2}$$

where W is the almost-surely winning region for the $\mathrm{Avg}_{>0}$ *game.*

Proof (Lemma 4). The winning condition $\mathrm{Avg}_{>0}$ is submixing and tail, hence there exists a positional optimal strategy [11]. Therefore, there exists a positional almost-surely winning strategy. Thus by Corollary 2, σ is almost-surely winning for $\underline{\mathrm{Avg}}_{>0}$ as well. Hence the following equation holds,

$$\forall v \in W,\ \mathbb{P}^{\sigma}_{v}\left(\liminf_{n} \frac{1}{n+1}\sum_{i=0}^{n} r(v_i) > 0\right) = 1 \ \ .$$

The same strategy σ yields (2). □

Proposition 3. *In any Markov decision process $\mathcal{M}$ equipped with* $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ *objective,* Player 0 *has an almost-surely winning strategy with finite memory.*

The main idea beyond Proposition 3 is that the strategy described in Section 4 can be decomposed into 2 phases.

Phase 1 when Player 0 is applying her attraction strategy π.
Phase 2 when Player 0 is applying her $\mathrm{Avg}_{>0}$ strategy τ.

After these two phases, Player 0 is ensured that her average payoff is strictly positive.

To prove Proposition 3, we show that by considering only the expected average reward, one can bound the time that Player 0 has to play each of the strategies of **Phase 1** and **Phase 2** and yet fulfill almost-surely the $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ objective.

Size of the Memory. The following proposition shows that an exponential memory is necessary and sufficient.

Proposition 4. *In a one-player Markov decision process $\mathcal{M}$ with the* $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ *objective, memory of size exponential in the size of the arena is both necessary and sufficient to implement an almost-surely winning strategies.*

Consider the Markov Decision Process depicted in Figure 1. We show that memory of size exponential in the size of the arena is necessary.

Player 0 wins almost-surely from any vertex $v \in \mathcal{M}$. Let σ be an almost-surely winning strategy with finite-memory of size k. Denote T_R the absorption time in vertex n in the Markov chain obtained by removing from $\mathcal{M}$ the edge $(n, 0)$. $\mathbb{E}^{\sigma}_{v}[T_R]$ gives the expected time to reach n if the initial state is v. Thus the expected average reward for Player 0 on the path from vertex 0 to vertex n is $-\mathbb{E}^{\sigma}_{v}[T_R]$. We show by contradiction that $k \geq \mathbb{E}^{\sigma}_{v}[T_R]$. Since σ is almost-surely winning, the strategy cannot cycle forever in vertex n thus it leaves vertex n after at most k loops. Thus the expected accumulated reward on a path from vertex 0 to itself is less than $k - \mathbb{E}^{\sigma}_{v}[T_R]$. If this value was negative then according to the law of large numbers, the expected average reward would be almost-surely negative as well, a contradiction. Since $\mathbb{E}^{\sigma}_{0}[T_R] \geq 2^{n+1}\left(1 - \frac{1}{2^n}\right)$ (proof in appendix), then σ has a memory at least exponential in the size of the arena.

Combining results of Proposition 3 and Proposition 4 lead Theorem 5 and thus the following corollary.

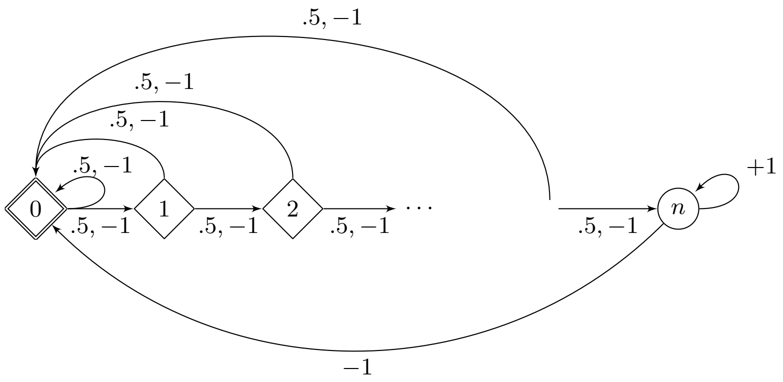

Fig. 1. Player 0 wins in $\mathcal{M}$ if she visits state 0 infinitely many times and she satisfies the $\mathrm{Avg}_{>0}$ objective.

Corollary 4. *In any Markov decision process $\mathcal{M}$ where the winning condition is* $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$, Player 0 *has an optimal strategy with memory of size exponential in the size of the arena.*

5 Solving Parity and Positive-average Games with lim inf semantics

In the previous section we studied parity and positive-average games with lim sup objective. An alternative definition of these games is to replace lim sup by lim inf. We show that all results of the previous section hold for this alternative definition.

Definition 15.

$$\underline{\mathrm{Avg}}_{>0} = \left\{ v_0 v_1 v_2 \cdots \in V^* \mid \liminf_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} r(v_i) > 0 \right\} \quad .$$

To compute the values in $\mathrm{Par} \wedge \underline{\mathrm{Avg}}_{>0}$ games, we use the previous result on optimality using finite memory and known results on Markov chains theory. Actually we show that the value of every vertex in the game $\mathrm{Par} \wedge \underline{\mathrm{Avg}}_{>0}$ is equal to the one in the game $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$.

Proposition 5. *In any Markov decision process $\mathcal{M}$ we have:*

$$\forall v \in V, \;\; \mathrm{val}_{\mathrm{Par} \wedge \mathrm{Avg}_{>0}}(v) = \mathrm{val}_{\mathrm{Par} \wedge \underline{\mathrm{Avg}}_{>0}}(v) \quad .$$

To prove this Proposition 5 we need the following lemma:

Lemma 5 (see e.g. [20]). *Let $\mathcal{M}$ be a finite Markov chain and $r : V \to \mathbb{R}$ a reward function. The following equality holds for almost all runs.*

$$\liminf_{n\to\infty} \sum_{i=0}^{n-1} \frac{r(v_i)}{n} = \limsup_{n\to\infty} \sum_{i=0}^{n-1} \frac{r(v_i)}{n} \quad .$$

Proof (Proposition 5). We show that the following inequalities holds:

$$\forall v \in V,\ \ \mathrm{val}_{\mathrm{Par}\wedge\underline{\mathrm{Avg}}_{>0}}(v) \leq \mathrm{val}_{\mathrm{Par}\wedge\mathrm{Avg}_{>0}}(v) \quad . \tag{3}$$

$$\forall v \in V,\ \ \mathrm{val}_{\mathrm{Par}\wedge\mathrm{Avg}_{>0}}(v) \leq \mathrm{val}_{\mathrm{Par}\wedge\underline{\mathrm{Avg}}_{>0}}(v) \quad . \tag{4}$$

That (3) holds is trivial. It is a consequence of the fact that every winning strategy for $\mathrm{Par}\wedge\underline{\mathrm{Avg}}_{>0}$ is also winning for $\mathrm{Par}\wedge\mathrm{Avg}_{>0}$.

To prove (4), notice that according to Corollary 4 Player 0 can play optimally using finite memory in the $\mathrm{Par}\wedge\mathrm{Avg}_{>0}$ game, thus there exists a strategy $\sigma^\sharp$ which is optimal and with finite memory. Hence:

$$\begin{aligned}
\mathrm{val}_{\mathrm{Par}\wedge\mathrm{Avg}_{>0}}(v) &= \mathbb{P}_v^{\sigma^\sharp}(\mathrm{Par}\wedge\mathrm{Avg}_{>0}) \\
&= \mathbb{P}_v^{\sigma^\sharp}(\mathrm{Par}\wedge\underline{\mathrm{Avg}}_{>0}) \\
&\leq \sup_\sigma \mathbb{P}_v^{\sigma}(\mathrm{Par}\wedge\underline{\mathrm{Avg}}_{>0}) = \mathrm{val}_{\mathrm{Par}\wedge\underline{\mathrm{Avg}}_{>0}}(v) \quad ,
\end{aligned}$$

where the first equality is by definition of the value and the second is by Lemma 5. Therefore (4) holds and Proposition 5 is proved. □

Proposition 5 leads the following theorem.

Theorem 6. *In any Markov decision process $\mathcal{M}$ where the winning condition is $\mathrm{Par}\wedge\underline{\mathrm{Avg}}_{>0}$, the values are computable in* polynomial *time. Moreover the optimal strategies can be implemented using finite memory.*

6 Conclusion

In this paper we have considered Markov decision processes equipped with parity and positive-average winning conditions. We have shown that finite memory optimal strategies exist and that a memory exponential in the size of the arena is sufficient to implement these strategies. We have also given a polynomial time algorithm to compute the values of such games. Moreover, our algorithm solves parity and positive-average games under both the lim sup and lim inf semantics.

References

1. Christel Baier, Frank Ciesinski, and Marcus Größer. Problem and verification of Markov decision processes. *SIGMETRICS Performance Evaluation Review*, 32(4):22–27, 2005.

2. Tomás Brázdil, Václav Brozek, and Kousha Etessami. One-counter stochastic games. In *FSTTCS*, pages 108–119, 2010.
3. Tomás Brázdil, Václav Brožek, and Kousha Etessami. One-counter stochastic games. *CoRR*, abs/1009.5636, 2010.
4. Krishnendu Chatterjee and Laurent Doyen. Energy parity games. In *ICALP (2)*, pages 599–610, 2010.
5. Krishnendu Chatterjee, Tom Henzinger, and Marcin Jurdzinski. Mean-payoff parity games. In *LICS 05*, June 2005.
6. Krishnendu Chatterjee, Marcin Jurdziński, and Thomas A. Henzinger. Quantitative stochastic parity games. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '04, pages 121–130, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.
7. C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events. In *ICALP'90*, volume 443 of *LNCS*, pages 336–349. Springer, 1990.
8. Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995.
9. E.A. Emerson and C.S. Jutla. Tree automata, mu-calculus and determinacy. *Foundations of Computer Science, Annual IEEE Symposium on*, 0:368–377, 1991.
10. Dean Gillette. Stochastic games with zero stop probability. *Contributions to the Theory of Games*, 3:179–187, 1957.
11. Hugo Gimbert. Pure stationary optimal strategies in Markov decision processes. In *STACS*, pages 200–211, 2007.
12. Hugo Gimbert and Florian Horn. Solving Simple Stochastic Tail Games. page 1000, 01 2010.
13. Hugo Gimbert and Wieslaw Zielonka. Deterministic priority mean-payoff games as limits of discounted games. In *ICALP (2)*, pages 312–323, 2006.
14. Hugo Gimbert and Wieslaw Zielonka. Limits of multi-discounted Markov decision processes. In *LICS*, pages 89–98, 2007.
15. E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics and Infinite Games*, volume 2500 of *LNCS*. Springer, 2002.
16. M. Vardi K. Etessami, M. Kwiatkowska and M. Yannakakis. Multi-objective model checking of markov decision processes. In *Proc of TACAS'07*, volume 4424, pages 50–65, 2007.
17. M. Kwiatkowska, G. Norman, and D. Parker. Stochastic model checking. In *Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation (SFM'07)*, 2007.
18. T. A. Liggett and S. A. Lippman. Stochastic games with perfect information and time average payoffs. *SIAM Review*, 11:604 – 607, 1969.
19. Martin L. Putterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, New York, NY, 1994.
20. J. R. Norris. *Markov chains*. Cambridge University Press, 1997.
21. Maria Mateescu Thomas A. Henzinger and Verena Wolf. Sliding-window abstraction for infinite markov chains. In *Proc. of CAV'09*, volume 5643, pages 337–352, 2009.
22. Wieslaw Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theor. Comput. Sci.*, 200(1-2):135–183, 1998.
23. Wieslaw Zielonka. Perfect-information stochastic parity games. In *FoSSaCS*, pages 499–513, 2004.

A Proof of Theorem 2

Proof. Since parity games are tail conditions, the first part of the theorem follows directly from Theorem 1.

Let $\mathcal{M}$ be a Markov decision process, To prove the second part of the theorem, notice that the parity condition can be written as disjoint union of winning condition where in each one, Player 0 wins if she satisfies the parity condition played in a parity game with three priorities. In other words, we define Φ_d as the game played in $\mathcal{M}$ where every vertex with even priority d is transformed into a vertex with priority 2, every vertex with priority greater than d is transformed to a vertex with priority 3 and every vertex with priority less than d is transformed into a vertex with priority 1. The game Φ_d can be solved in polynomial time since the number of priorities is fixed. Since Player 0 wins the original parity game if and only if she wins one of the games Φ_d for some even priority d, we get by Proposition 2 that the almost-surely winning region for the original parity game $W = \mathrm{Attr}_{=1}(\bigcup_{d \in D} W_{=1}[\phi_d])$ where D is the set of all even priorities. Corollary 1 shows that the values can be computed in polynomial time and thus finishes the proof. □

B Proof of Lemma 1

Proof. To prove the lemma we show the following:

(i) Any set $X \subseteq V$ satisfying 1, 2 and 3 is almost-surely winning.
(ii) The almost surely winning region satisfies 1, 2 and 3.

We start by proving (i). Let $X \subseteq V$, we exhibit an almost-surely winning strategy σ for Player 0 from any vertex in X. Strategy σ is defined as follows. If the play ever reaches $X \setminus \mathrm{Attr}_{>0}(V_d \cap X, X)$, then Player 0 forgets everything that happened up to now and starts playing an almost-surely winning strategy $\sigma_{=1}$ in $\mathcal{A}[X \setminus \mathrm{Attr}_{>0}(V_d \cap X, X)]$, which exists by (3). As long as the play stays in $\mathrm{Attr}_{>0}(V_d \cap X, X)$, strategy σ is defined as follows. Initially, she plays her attractor strategy to $V_d \cap X$, π for $|X|$ steps. Then she plays her almost-surely winning strategy for the $\mathrm{Avg}_{>0}$ objective, τ until her average payoff goes above an appropriately chosen $\eta > 0$. Then she starts from scratch.

We now prove that the strategy σ described above is almost-surely winning. If ever a play consistent with σ reaches $X \setminus \mathrm{Attr}_{>0}(V_d, X)$, then according to Proposition 1 it will stay trapped in $X \setminus \mathrm{Attr}_{>0}(V_d, X)$. Thus any play consistent with σ is almost-surely winning.

Assume now that the play stays is in $\mathrm{Attr}_{>0}(V_d, X)$. First, we show that the parity objective is satisfied. Let A_n be the event: *"A vertex with priority d is not visited within n back and forth switches between τ and π"*. Let q be the minimum of the probabilities of all the chance edges in the arena $\mathcal{A}$. We have for every $v \in \mathrm{Attr}_{>0}(V_d, X)$ $\mathbb{P}_v^\sigma(A_n) \leq (1 - q^{|X|})^n \leq (1 - q^{|V|})^n$. Now, because $(1 - q^{|V|}) < 1$, $\sum_{n>0} \mathbb{P}_v^\sigma(A_n) \leq \sum_{n>0} (1 - q^{|V|})^n < \infty$. Thus using Borel-Cantelli Lemma we get $\mathbb{P}_v^\sigma(\bigcap_{n=1}^\infty \bigcup_{k=n}^\infty A_k) = 0$. That is, the probability that infinitely many of the events A_n happen is 0. Hence a vertex with priority d is eventually visited, and the parity objective is almost-surely satisfied.

Next, we prove that the positive-average objective is satisfied. Notice that since τ is positional, it also achieves the positive-average condition with the $\liminf$ semantics. Thus there exists an integer η such that the accumulated average reward eventually never goes under $\eta > 0$. This is feasible since the play is happening in the almost-surely winning region for the positive-average condition. After reaching a vertex of priority d or applying the attractor strategy for $|X|$ steps, Player 0 applies her strategy τ until her average payoff goes above η. Thus the $\mathrm{Avg}_{>0}$ objective is achieved almost-surely. The above facts show that σ is almost-surely winning. This completes the proof of (i). Note also that this strategy may require an infinite memory size since the time needed by Player 0 to make her average payoff greater than η is not bounded.

We now show (ii). Denote by W the almost-surely winning region. We prove that W satisfies 1, 2 and 3. That 1 holds is obvious. That W satisfies 2 follows from the fact that Player 0 can ensure almost-surely the conjunction $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ in $\mathcal{A}[W]$. To see that 3 holds, note that $G[W \setminus \mathrm{Attr}_{>0}(V_d, W)]$ is a trap for Player 0. So if she plays her almost-surely winning strategy σ defined on W, she wins almost surely the $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ condition. This shows (ii). □

C Proof of Lemma 2

Proof. We show that from any vertex in the described set, Player 0 has an almost-surely winning strategy. Player 0 applies the following strategy. As long as the game has not reached R, player 0 plays her attractor strategy π. If the play is in R, she uses her almost-surely winning strategy, τ, in R. That is,

$$\sigma : V \longrightarrow V$$

$$\sigma(v) = \begin{cases} \pi(v) \text{ if } v \notin R \\ \tau(v) \text{ if } v \in R. \end{cases}$$

This strategy is almost-surely winning since any play consistent with it eventually reaches the set R and stays there forever.

We now prove that the almost-surely winning region is exactly the set described by the lemma. For this, let W be a set of vertices satisfying the claim of the lemma. We show that Player 0 cannot win almost-surely in $V \setminus W$. Let σ' a strategy and $v \in V \setminus W$. Any play consistent with σ' either a) visits $V \setminus \mathrm{Safe}(V_d, V)$ infinitely often or b) the play ultimately reaches $\mathrm{Safe}(V_d, V)$. If case a) holds, using the Borel-Cantelli Lemma we get that a vertex of priority d is visited infinitely many times. If b) holds, let A be the random variable with values in $|V|$ which denotes the first vertex reached in the set $\mathrm{Safe}(V_d, V)$. If $\mathbb{P}_A^{\sigma'}(\mathrm{Par} \wedge \mathrm{Avg}_{>0}) = 1$, it would mean that $A \in W$. Hence $\mathbb{P}_A^{\sigma'}(\mathrm{Par} \wedge \mathrm{Avg}_{>0}) < 1$ which implies $\mathbb{P}_v^{\sigma'}(\neg(\mathrm{Par} \wedge \mathrm{Avg}_{>0})) \geq \mathbb{P}_A^{\sigma'}(\neg(\mathrm{Par} \wedge \mathrm{Avg}_{>0})) > 0$. This shows that the Player 0 cannot win almost-surely from any vertex in $V \setminus W$. □

D Proof of Lemma 3

Proof. Let $\mathcal{M}$ be a finite irreducible Markov chain with reward. Suppose that the total reward of $\mathcal{M}$ diverge to infinity. Let s a recurrent state and T_s the random variable that gives the time to the next visit of s when starting from s. We are interested in the following quantity $\mathbb{E}_s\left[\frac{1}{T_s}\sum_{i=0}^{T_s-1} r(v_i)\right]$. According to [3] (Lemma 11) this quantity is strictly positive if and only if the accumulated reward of $\mathcal{M}$ diverges to infinity. This proves the first part of the lemma.

We use a discounted approximation to compute a lower bound η. Let $0 < \lambda < 1$ and V_λ a vector such that for every state s of $\mathcal{M}$, $V_\lambda(s) = \mathbb{E}_s[(1-\lambda)\sum_{i\leq 0}\lambda^i r(v_i)]$. It is well known that $a)\lim_{\lambda\to 1} V_\lambda(s) = \mathbb{E}_s[\frac{1}{n}\sum_{i\leq 0} r(v_i)]$. This relation holds even in the more general case of simple stochastic games [10, 18]. $b)V_\lambda = (1-\lambda)R + \lambda P V_\lambda$, where R is the reward vector and P is the transition matrix of $\mathcal{M}$. Hence, by b) we have that $V_\lambda = (1-\lambda P)^{-1}(1-\lambda)R$, thus $V_\lambda(s) = (1-\lambda P)^{-1}(1-\lambda)R(s)$. This quantity is a rational fraction of λ, therefore there exists two polynomials Q and S with degree at most n, where n the number of states of $\mathcal{M}$ such that $V_\lambda(s) = \frac{Q(\lambda)}{S(\lambda)}$. According to a) we can write $\lim_{\lambda\to 1} V_\lambda(s) = \frac{Q(1)}{S(1)}$, this quantity has a polynomial-bit complexity. Thus, there is a polynomial P such that $V_\lambda \geq 2^{-P(n)}$. Using the strong Markov property we have $\lim_{\lambda\to 1} V_\lambda(s) = \mathbb{E}_s[\frac{1}{t_s}\sum_{i=0}^{t_s-1} r(v_i)]$. □

E Proof of Proposition 3

Proof. Let $\mathcal{M}$ be a Markov decision process. We prove by induction on the number of priorities that Player 0 has an almost-surely winning strategies with finite memory. Suppose that $\mathcal{M}$ has one priority c. If c is even then Player 0 plays a positive average game, according to Theorem 3, there exist a positional optimal strategy for her. If c is odd then Player 0 has no winning strategy.

Suppose that Player 0 can win almost-surely using finite memory in any Markov decision process which contains less than d priority. Let $\mathcal{M}$ be a Markov decision process with d priorities.

If d is *odd*, according to Lemma 2, to win Player 0 applies her attractor strategy until she reaches the almost-surely winning region for the game $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ played in the subarena $\mathcal{A}[\mathrm{Safe}(V_d, V)]$. Note that in this subarena the number of priorities is strictly less than d and thus she has a finite memory strategy. Since the attraction strategy is memoryless, Player 0 has a finite memory strategy to win almost-surely if the highest priority is odd.

If d is *even*, according to Lemma 1, either Player 0 is playing in the almost-surely winning region for $\mathrm{Par} \wedge \mathrm{Avg}_{>0}$ in the subarena $\mathcal{A}[V \setminus \mathrm{Attr}^0_{>0}(V_d, V)]$ or she is playing outside that arena. In the former case, by induction, Player 0 has a finite memory strategy to win and the proof is done. In the latter case she applies her attractor strategy π for a specified time, then she switches to her positive-average strategy τ to ensure this objective as well. In the remaining of this proof, we are going to show that the time Player 0 should apply τ so that the positive-average objective is achieved can be bounded.

While being consistent with τ, Player 0 eventually reaches a recurrent state in the Markov chain $\mathcal{M}[\tau]$, denote this state u. Since τ is almost-surely winning for the $\mathrm{Avg}_{>0}$ objective, Lemma 4 shows us that the total reward of $\mathcal{M}$ will diverge to ∞. Lemma 3 on the other hand, shows that the expected average reward accumulated between two consecutive visits of u is strictly positive. Let η be the lowest expected average reward accumulated between two consecutive hits of a recurrent state in $\mathcal{M}[\tau]$ by Lemma 3 we know that $\eta > 0$. We recall also that the strategy that we are constructing does not differ much from the one of Lemma 1, where Player 0 plays in turn the strategy π and τ. Denote $n_0 \leq n_1 \leq \cdots$ the moments when Player 0 starts from scratch. To win almost-surely Player 0 has to ensure the following invariant throughout the play.

$$\forall s \in V,\ \forall i,\ \mathbb{E}_s^\sigma \left[\frac{1}{n_{i+1} - n_i} \sum_{k=n_i}^{n_{i+1}-1} r(v_k) \right] \geq x > 0\ . \tag{5}$$

We show that if (5) holds then

$$\forall s \in V,\ \mathbb{E}_s^\sigma \left[\limsup_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} r(v_k) \right] > 0\ .$$

Let $x > 0$, then for every vertex v we have

$$\begin{aligned} \forall i,\ \mathbb{E}_s^\sigma \left[\frac{1}{n_{i+1} - n_i} \sum_{k=n_i}^{n_{i+1}-1} r(v_k) \right] \geq x &\Rightarrow \forall i,\ \mathbb{E}_s^\sigma \left[\frac{1}{n_i} \sum_{k=0}^{n_i-1} r(v_k) \right] \geq x \\ &\Rightarrow \limsup_{n\to\infty} \mathbb{E}_s^\sigma \left[\frac{1}{n} \sum_{k=0}^{n-1} r(v_k) \right] \geq x \qquad (6) \\ &\Rightarrow \mathbb{E}_s^\sigma \left[\limsup_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} r(v_k) \right] \geq x\ . \qquad (7) \end{aligned}$$

Where the trasformation from (7) to (7) is by Fatou lemma. Hence by using the strong law of large numbers we get that the positive-average objective is ensured almost-surely.

After applying π for $|V|$ steps, Player 0 starts applying τ and increments the counter i each time the recurrent state R is visited. Let n the number of visits such that Equation 5 holds. For each even priority d we need the following memory $M_d = V \times \{0,1,2\} \times \{0,\cdots,|V|-1\} \times \{0,\cdots,n\}$. Let $\mathrm{Update} : V \times M_d \to M_d$ be the update function such that,

$$(u, v, b, i, j) = \begin{cases} (v, 0, i, j+1) \text{ if } (b = 0) \wedge (j < |V| - 1) \wedge (\chi(u) \neq d) \ . \\ (v, 1, i, j) \text{ if } (b = 0) \wedge [(j = |V| - 1) \vee (\chi(u) = d)] \ . \\ (v, 1, i, j) \text{ if } (b = 1) \wedge (u \notin R) \ . \\ (u, 2, 0, j) \text{ if } (b = 1) \wedge (u \in R) \ . \\ (v, 2, i+1, j) \text{ if } (b = 2) \wedge (u = v) \wedge (i < n) \ . \\ (v, 2, i, j) \text{ if } (b = 2) \wedge (u \neq v) \wedge (i < n) \ . \\ (v, 0, i, 0) \text{ if } (i = n) \ . \end{cases}$$

The strategy $\sigma : V \times M_d \to V$ consists in applying π the attractor strategy whenever $b = 0$ and applying τ the $\mathrm{Avg}_{>0}$ strategy whenever $b \neq 0$. □

F Proof of Proposition 4

Proof. We start with proving that exponential memory is sufficient. As in the proof of Proposition 3, let $\mathcal{M}$ a Markov decision process and by $\mathcal{M}[\tau]$, $\mathcal{M}[\pi]$ denote the Markov chains induced by τ and π respectively. We define the following random variables,

- T_R: with values in $\mathbb{N}$, is the time to absorption in $\mathcal{M}[\tau]$.
- T_n: with values in $\mathbb{N}$, is the time needed to visit the first recurrent state reached $n+1$ times.

Note that if all the rewards in $\mathcal{M}$ are strictly positive, Player 0 plays only for the parity objective. Hence no memory is required.

Now assume that theres exist non positive rewards in $\mathcal{M}$. We want to compute an integer n such that Equation (5) in the proof of Proposition 3 holds.

$$\frac{1}{T_n} \sum_{i=0}^{T_n - 1} r(v_i) = \frac{1}{T_n} \left[\sum_{i=0}^{|V|-1} r(v_i) + \sum_{i=|V|}^{T_0 - 1} r(v_i) + \sum_{j=0}^{n-1} \sum_{i=T_j}^{T_{j+1}-1} r(v_i) \right] \ .$$

Let

- $a = \sum_{i=0}^{|V|-1} r(v_i)$.
- $b = \sum_{i=|V|}^{T_0 - 1} r(v_i)$.
- $c_j = \sum_{i=T_j}^{T_{j+1}-1} r(v_i)$.

Hence for every $s \in V$

$$\mathbb{E}_s^\sigma \left[\frac{1}{T_n} \sum_{i=0}^{T_n - 1} r(v_i) \right] = \mathbb{E}_s^\sigma \left[\frac{a}{T_n} + \frac{b}{T_n} + \frac{\sum_{j=0}^{n-1} c_j}{T_n} \right] \ .$$

We first compute a lower bound for $\mathbb{E}_s^\sigma\left[\frac{a}{T_n}\right]$.

$$\frac{1}{T_n}\sum_{i=0}^{|V|-1} r(v_i) = \frac{|V|}{T_n}\frac{1}{|V|}\sum_{i=0}^{|V|-1} r(v_i) \geq \frac{|V|}{n}\min_{v\in V}\{r(v)\}\ .$$

Where the inequality holds because $T_n \geq n$ and $\min_{v\in V}\{r(v)\}$ is negative. Hence

$$\mathbb{E}_s^\sigma\left[\frac{a}{T_n}\right] \geq \frac{|V|}{n}\min_{v\in V}\{r(v)\}\ . \tag{8}$$

Next, we compute a lower bound for $\mathbb{E}_s^\sigma\left[\frac{b}{T_n}\right]$

$$\begin{aligned}
\mathbb{E}_s^\sigma\left[\frac{1}{T_n}\sum_{i=|V|}^{T_0-1} r(v_i)\right] &= \mathbb{E}_s^\sigma\left[\frac{T_0-|V|}{T_n}\frac{1}{T_0-|V|}\sum_{i=|V|}^{T_0-1} r(v_i)\right]\\
&\geq \mathbb{E}_s^\sigma\left[\frac{T_0-|V|}{T_n}\min_{v\in|V|}\{r(v)\}\right]\\
&\geq \mathbb{E}_s^\sigma\left[\frac{T_0-|V|}{n}\min_{v\in|V|}\{r(v)\}\right] = \frac{\mathbb{E}_s^\sigma\left[T_0-|V|\right]}{n}\min_{v\in|V|}\{r(v)\}
\end{aligned}$$

Where the first inequality holds because $T_n \geq n$ and $\min_{v\in V}\{r(v)\}$ is negative. Hence

$$\mathbb{E}_s^\sigma\left[\frac{b}{T_n}\right] \geq \frac{\mathbb{E}_s^\sigma\left[T_0-|V|\right]}{n}\min_{v\in|V|}\{r(v)\}\ . \tag{9}$$

Finally, we compute a lower bound for $\mathbb{E}_s^\sigma\left[\frac{\sum_{j=0}^{n-1} c_j}{T_n}\right]$.

$$\begin{align}
\mathbb{E}_s^\sigma\left[\frac{1}{T_n}\sum_{i=T_0}^{T_n-1} r(v_i)\right] &= \mathbb{E}_s^\sigma\left[\sum_{j=0}^{n-1}\frac{1}{T_n}\sum_{i=T_j}^{T_{j+1}-1} r(v_i)\right] \notag\\
&= \mathbb{E}_s^\sigma\left[\sum_{j=0}^{n-1}\frac{T_{j+1}-T_j}{T_n}\frac{1}{T_{j+1}-T_j}\sum_{i=T_j}^{T_{j+1}-1} r(v_i)\right] \notag\\
&\geq \mathbb{E}_s^\sigma\left[\sum_{j=0}^{n-1}\frac{T_{j+1}-T_j}{T_n}\mathbb{E}_s^\sigma\left[\frac{1}{T_{j+1}-T_j}\sum_{i=T_j}^{T_{j+1}-1} r(v_i)\ \middle|\ T_j\right]\right] \notag\\
&= \mathbb{E}_s^\sigma\left[\frac{T_n-T_0}{T_n}\mathbb{E}_s^\sigma\left[\frac{1}{T_1-T_0}\sum_{i=T_0}^{T_1-1} r(v_i)\right]\right] \tag{10}\\
&\geq \eta\mathbb{E}_s^\sigma\left[1-\frac{T_0}{T_n}\right] \tag{11}\\
&\geq \eta\left(1-\frac{\mathbb{E}_s^\sigma\left[T_0\right]}{n}\right) \tag{12}\\
&\geq \eta\left(1-\frac{\mathbb{E}_s^\sigma\left[T_0-|V|\right]+|V|}{n}\right) \notag
\end{align}$$

Where the transformation from (10) to (11) holds by Lemma 3 and from (11) to (12) because $T_n \geq n$. Hence,

$$\mathbb{E}_s^{\tau}\left[\frac{\sum_{j=0}^{n-1} c_j}{T_n}\right] \geq \eta \left(1 - \frac{\mathbb{E}_s^{\sigma}\left[T_0 - |V|\right] + |V|}{n}\right) \quad . \tag{13}$$

From (8), (9) and (13) we get

$$\mathbb{E}_s^{\sigma_n}\left[\frac{1}{T_n}\sum_{i=0}^{T_n-1} r(v_i)\right] \geq \frac{|V|}{n} m + \frac{\mathbb{E}_s^{\sigma}\left[T_R\right]}{n} m + \eta\left(1 - \frac{\mathbb{E}_s^{\sigma}\left[T_R\right] + |V|}{n}\right)$$

Let us find a value for n such that

$$\frac{m}{n}\left(|V| + \mathbb{E}_s^{\sigma}\left[T_R\right]\right) + \frac{\eta}{n}\left(n - \mathbb{E}_s^{\sigma}\left[T_R\right] + |V|\right) > 0$$

We find

$$\begin{aligned} n &> \mathbb{E}_s^{\sigma}[T_R] + |V| - \frac{m}{\eta}\left(|V| + \mathbb{E}_s^{\sigma}[T_R]\right) \\ &\geq \mathbb{E}_s^{\sigma}[T_R] + |V| - m2^{Q(|V|)}\left(|V| + \mathbb{E}_s^{\sigma}[T_R]\right) \end{aligned}$$

We compute an upper bound for $\mathbb{E}_s^{\sigma}[T_R]$. Let P, the sub-stochastic matrix obtained from the transition matrix of $\mathcal{M}[\tau]$ by replacing every recurrent entry by 0. From any state v in $\mathcal{M}[\tau]$, the time to absorption is given by $(I-P)^{-1}(v)$. Hence

$$\mathbb{E}_s^{\sigma}[T_R] \leq |V| + \max_{v \in |V|}\{(I-P)^{-1}(v)\} \quad .$$

Using the same arguments as in Lemma 3, we get that this quantity is exponential in a polynomial in the size of the arena.

The proof that exponential size memory is necessary is given after Proposition 4 except for the lower bound on absorption time. This lower bound is established in Proposition 6.

□

G Details of Computations of Figure 1

Proposition 6. *In the game depicted on Figure 1, the expected absorption time from vertex* 0 *to vertex* n *is at least*

$$2^{n+1}\left(1 - \frac{1}{2^n}\right) .$$

Proof. We compute a lower bound for $\mathbb{E}_v^\sigma[T_R]$.

We use a first step analysis to compute $\mathbb{E}_0^\sigma[T_R]$. We know that for every state $0 \leq i \leq n-1$

$$\mathbb{E}_i^\sigma[T_R] = 1 + \frac{1}{2}\mathbb{E}_0^\sigma[T_R] + \frac{1}{2}\mathbb{E}_{i+1}^\sigma[T_R] \ \ .$$

and for $i = n$

$$\mathbb{E}_n^\sigma[T_R] = 0 \ \ .$$

Thus we get

$$\mathbb{E}_0^\sigma[T_R] = 2^n \sum_{i=0}^{n-1} \frac{1}{2^i} = 2^{n+1}\left(1 - \frac{1}{2^n}\right) \ \ .$$

□