Meet Your Expectations With Guarantees: Beyond Worst-Case Synthesis in Quantitative Games*

Véronique Bruyère[1], Emmanuel Filiot[2,3,†], Mickael Randour[1,‡], and Jean-François Raskin[3,§]

[1] Computer Science Department, Université de Mons (UMONS), Belgium
[2] LACL, Paris-Est Créteil, France
[3] Département d'Informatique, Université Libre de Bruxelles (U.L.B.), Belgium

Abstract. We extend the quantitative synthesis framework by going beyond the worst-case. On the one hand, classical analysis of two-player games involves an adversary (modeling the environment of the system) which is purely antagonistic and asks for strict guarantees. On the other hand, stochastic models like Markov decision processes represent situations where the system is faced to a purely randomized environment: the aim is then to optimize the expected payoff, with no guarantee on individual outcomes. We introduce the beyond worst-case synthesis problem, which is to construct strategies that guarantee some quantitative requirement in the worst-case while providing an higher expected value against a particular stochastic model of the environment given as input. This problem is relevant to produce system controllers that provide nice expected performance in the everyday situation while ensuring a strict (but relaxed) performance threshold even in the event of very bad (while unlikely) circumstances. We study the beyond worst-case synthesis problem for two important quantitative settings: the mean-payoff and the shortest path. In both cases, we show how to decide the existence of finite-memory strategies satisfying the problem and how to synthesize one if one exists. We establish algorithms and we study complexity bounds and memory requirements.

1 Introduction

Two-player zero-sum quantitative games [13,30,3] and Markov decision processes (MDPs) [26,4] are two popular formalisms for modeling decision making in adversarial and uncertain environments respectively. In the former, two players compete with opposite goals (zero-sum), and we want strategies for player 1 (the system) that ensure a given *minimal performance against all possible strategies* of player 2 (its environment). In the latter, the system plays against a stochastic model of its environment, and we want strategies that ensure a *good expected overall performance*. Those two models are well studied and simple optimal memoryless strategies exist for classical objectives such as mean-payoff [24,13,14] or shortest path [2,11]. But both models have clear weaknesses: strategies that are good for the worst-case may exhibit suboptimal behaviors in probable situations while strategies that are good for the expectation may be terrible in some unlikely but possible situations.

In practice, we would like to have strategies that are both ensuring (a) some worst-case threshold no matter how the adversary behaves (i.e., against any arbitrary strategy) and (b) a good expectation against the expected behavior of the adversary (given as a stochastic model). This is the subject of this paper: we show how to construct finite-memory strategies that ensure both (a) and (b). We consider finite-memory strategies for player 1 as they can be implemented in practice (as opposed to infinite-memory ones). Player 2 is not restricted in his choice of strategies, but we will see that simple strategies suffice. Our problem, the **beyond worst-case synthesis problem**, is interesting for any quantitative measure, but we give here a thorough study of two classical ones: the *mean-payoff*, and the *shortest path*.

Example. Let us consider the weighted game in Fig. 1 to illustrate the *shortest path* context. Circle states belong to player 1, square states to player 2, integer labels are durations in minutes, and fractions are probabilities that model the expected behavior of player 2. Player 1 wants a strategy to go from "home" to "work" such that "work" is *guaranteed* to be reached within 60 minutes (to avoid missing an important meeting), and player 1 would also like to minimize the expected time to reach "work". First, note that the strategy that minimizes the expectation is to take the car (expectation

* Work partially supported by European project CASSTING (FP7-ICT-601148).
† Chercheur qualifié F.R.S.-FNRS.
‡ Author supported by F.R.S.-FNRS fellowship.
§ Author supported by ERC Starting Grant (279499: inVEST).

is 33 minutes) but this strategy is excluded as there is a possibility to arrive after 60 minutes (in case of heavy traffic). Bicycle is safe but the expectation of this solution is 45 minutes. We can do better with the following strategy: try to take the train, if the train is delayed three time consecutively, then go back home and take the bicycle. This strategy is safe as it always reach "work" within 59 minutes and its expectation is $\approx 37,56$ minutes (so better than taking directly the bicycle). Our algorithms are able to decide the existence of (and synthesize) such finite-memory strategies.

Contributions. Our main results are the following. First, for the mean-payoff value, we provide an NP $\cap$ coNP algorithm (Thm. 1), which would be in P if mean-payoff games were proved to be in P, a long-standing open problem [3,6]. Pseudo-polynomial memory may be necessary and always suffices (Thm. 4). Finally, we observe that infinite-memory strategies are strictly more powerful that finite-memory strategies (Sect. 4.11). Second, for the shortest path, we provide a pseudo-polynomial time algorithm (Thm. 5), and show that the associated decision problem is NP-hard (Thm. 7). Pseudo-polynomial memory may be necessary and always suffices (Thm. 6). In the case of the shortest path problem, infinite-memory strategies grant no additional power in comparison with finite-memory strategies (Rem. 10).

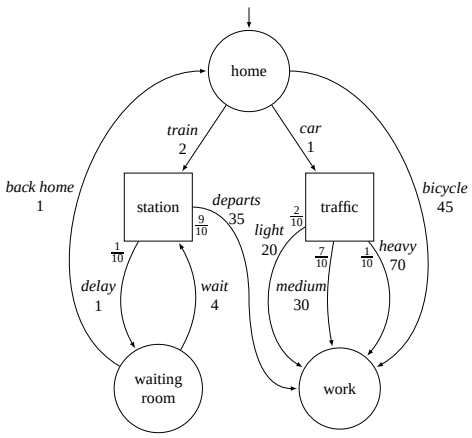

Fig. 1: Player 1 wants to minimize its expected time to reach "work", but while ensuring it is less than an hour in all cases.

Related works. Our problems generalize the corresponding problems for two-player zero-sum games and MDPs. In mean-payoff games, optimal memoryless worst-case strategies exist and the best known algorithm is in NP $\cap$ coNP [13,30,3]. For shortest path games,[4] it can be shown that memoryless strategies also suffice, and the problem is in P. In MDPs, optimal strategies for the expectation are studied in [26,14] for the mean-payoff and the shortest path: in both cases, memoryless strategies suffice and they can be computed in P.

Our strategies are *strongly risk averse*: they avoid at all cost outcomes that are below a given threshold (no matter what is their probability), and inside the set of those *safe* strategies, we maximize expectation. To the best of our knowledge, we are the first to consider such strategies. Other different notions of risk have been studied for MDPs: for example in [29], the authors want to find policies which minimize the probability (risk) that the total discounted rewards do not exceed a specified value (target), or in [15] the authors want policies that achieve a specified value of the long-run limiting average reward at a specified probability level (percentile). While those strategies limit risk, they only ensure *low probability* for bad behaviors but they do not ensure their absence, furthermore, they do not ensure good expectation either.

Structure of the paper. In Sect. 2, we introduce the necessary definitions. In Sect. 3, we formally define the beyond worst-case synthesis problem. Sect. 4 and Sect. 5 are respectively devoted to the solutions for the mean-payoff and the shortest path. We conclude (Sect. 6) with a comparative note on the two solutions.

2 Preliminaries

Weighted directed graphs. A *weighted directed graph* is a tuple $\mathcal{G} = (S, E, w)$ where (i) S is the set of vertices, called *states*; (ii) $E \subseteq S \times S$ is the set of directed edges; and (iii) $w\colon E \to \mathbb{Z}$ is the weight labeling function. Since we only work with directed graphs in the following, we omit the adjective and talk about *weighted graphs*. Also, in the sequel, we almost exclusively work with *finite* graphs, i.e., graphs for which the set of states S is finite. Given a state $s \in S$, we denote by $\mathsf{Succ}(s) = \{s' \in S \mid (s,s') \in E\}$ the set of successors of s by edges in E. We assume that graphs are non-blocking, i.e., for all $s \in S$, $\mathsf{Succ}(s) \neq \emptyset$. We denote by W the largest absolute weight that appears in the graph. We assume that weights are encoded in binary and denote by $V = \lceil \log_2 W \rceil$ the number of bits of their encoding.

A *play* in $\mathcal{G}$ from an initial state $s_{\mathsf{init}} \in S$ is an infinite sequence of states $\pi = s_0 s_1 s_2 \dots$ such that $s_0 = s_{\mathsf{init}}$ and $(s_i, s_{i+1}) \in E$ for all $i \geq 0$. The *prefix* up to the n-th state of π is the finite sequence $\pi(n) = s_0 s_1 \dots s_n$. We resp. denote

[4] We consider game graphs with strictly positive weights ant try to minimize the cost to target.

the first and last states of the prefix by $\mathsf{First}(\pi(n)) = s_0$ and $\mathsf{Last}(\pi(n)) = s_n$. For a play π, we naturally extend the notation to $\mathsf{First}(\pi)$. The set of plays of $\mathcal{G}$ is denoted by $\mathsf{Plays}(\mathcal{G})$ and the corresponding set of prefixes is denoted by $\mathsf{Prefs}(\mathcal{G})$. Given a play $\pi \in \mathsf{Plays}(\mathcal{G})$, we denote by $\mathsf{Inf}(\pi) \subseteq S$ the set of states that are visited infinitely often along the play.

Given a function $f\colon \mathsf{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$, the *value* of a play π is denoted $f(\pi)$. We consider two classical value functions, the *total-payoff* and the *mean-payoff*, defined as follows. The *total-payoff* of a prefix $\rho = s_0 s_1 \dots s_n$ is $\mathsf{TP}(\rho) = \sum_{i=0}^{i=n-1} w((s_i, s_{i+1}))$, and its *mean-payoff* is $\mathsf{MP}(\rho) = \frac{1}{n}\mathsf{TP}(\rho)$. This is naturally extended to plays by considering the limit behavior: the total-payoff of a play π is $\mathsf{TP}(\pi) = \liminf_{n\to\infty} \mathsf{TP}(\pi(n))$ and its mean-payoff is $\mathsf{MP}(\pi) = \liminf_{n\to\infty} \mathsf{MP}(\pi(n))$. Given a graph $\mathcal{G}$ where all weights are strictly positive (i.e., $w\colon E \to \mathbb{N}_0$) and a target set of states $T \subseteq S$, we define the *truncated sum up to* T as $\mathsf{TS}_T\colon \mathsf{Plays}(\mathcal{G}) \to \mathbb{N} \cup \{\infty\}$, $\mathsf{TS}_T(\pi = s_0 s_1 s_2 \dots) = \sum_{i=0}^{n-1} w((s_i, s_{i+1}))$, with n the first index such that $s_n \in T$, and $\mathsf{TS}_T(\pi) = \infty$ if π never reaches any state in T. As all weights are strictly positive, it is possible to reduce the truncated sum to the total-payoff (i.e., for all $\pi \in \mathsf{Plays}(\mathcal{G})$, $\mathsf{TS}_T(\pi) = \mathsf{TP}(\pi)$) by making all states of T absorbing with a self-loop of zero weight. That is, for all $s \in T$, we have that $\mathsf{Succ}(s) = \{s\}$ and $w((s,s)) = 0$.

Probability distributions. Given a finite set A, a (rational) *probability distribution* on A is a function $p\colon A \to [0,1] \cap \mathbb{Q}$ such that $\sum_{a\in A} p(a) = 1$. We denote the set of probability distributions on A by $\mathcal{D}(A)$. The *support* of the probability distribution p on A is $\mathsf{Supp}(p) = \{a \in A \mid p(a) > 0\}$.

Two-player games. We consider two-player turn-based games and denote the two *players* by $\mathcal{P}_1$ and $\mathcal{P}_2$. A finite *two-player game* is a tuple $G = (\mathcal{G}, S_1, S_2)$ composed of (i) a finite weighted graph $\mathcal{G} = (S, E, w)$; and (ii) a partition of its states S into S_1 and S_2 that resp. denote the sets of states belonging to $\mathcal{P}_1$ and $\mathcal{P}_2$. A prefix $\pi(n)$ of a play π belongs to $\mathcal{P}_i$, $i \in \{1,2\}$, if $\mathsf{Last}(\pi(n)) \in S_i$. The set of prefixes that belong to $\mathcal{P}_i$ is denoted by $\mathsf{Prefs}_i(G)$. We sometimes denote by $|G|$ the size of a game, defined as a polynomial function of $|S|$, $|E|$ and $V = \lceil \log_2 W \rceil$.

Strategies. Let $G = (\mathcal{G}, S_1, S_2)$ be a two-player game, a *strategy* for $\mathcal{P}_i$, $i \in \{1,2\}$, is a function $\lambda_i\colon \mathsf{Prefs}_i(G) \to \mathcal{D}(S)$ such that for all $\rho \in \mathsf{Prefs}_i(G)$, we have $\mathsf{Supp}(\lambda_i(\rho)) \subseteq \mathsf{Succ}(\mathsf{Last}(\rho))$. A strategy is called *pure* if it is deterministic, i.e., if its support is a singleton for all prefixes. When a strategy λ_i of $\mathcal{P}_i$ is pure, we sometimes simplify its notation and write $\lambda_i(\rho) = s$ instead of $\lambda_i(\rho)(s) = 1$, for any $\rho \in \mathsf{Prefs}_i(G)$ and the unique state $s \in \mathsf{Supp}(\lambda_i(\rho))$.

A strategy λ_i for $\mathcal{P}_i$ has *finite memory* if it can be encoded by a stochastic finite state machine with outputs, called *stochastic Moore machine*, $\mathcal{M}(\lambda_i) = (\mathsf{Mem}, \mathsf{m}_0, \alpha_\mathsf{u}, \alpha_\mathsf{n})$, where (i) Mem is a finite set of memory elements, (ii) $\mathsf{m}_0 \in \mathsf{Mem}$ is the initial memory element, (iii) $\alpha_\mathsf{u}\colon \mathsf{Mem} \times S \to \mathsf{Mem}$ is the update function, and (iv) $\alpha_\mathsf{n}\colon \mathsf{Mem} \times S_i \to \mathcal{D}(S)$ is the next-action function. If the game is in $s \in S_i$ and $\mathsf{m} \in \mathsf{Mem}$ is the current memory element, then the strategy chooses s', the next state of the game, according to the probability distribution $\alpha_\mathsf{n}(\mathsf{m}, s)$. When the game leaves a state $s \in S$, the memory is updated to $\alpha_\mathsf{u}(\mathsf{m}, s)$. Formally, $(\mathsf{Mem}, \mathsf{m}_0, \alpha_\mathsf{u}, \alpha_\mathsf{n})$ defines the strategy λ_i such that $\lambda_i(\rho \cdot s) = \alpha_\mathsf{n}(\hat{\alpha}_\mathsf{u}(\mathsf{m}_0, \rho), s)$ for all $\rho \in \mathsf{Prefs}(\mathcal{G})$ and $s \in S_i$, where $\hat{\alpha}_\mathsf{u}$ extends α_u to sequences of states starting from m_0 as expected. Note that pure finite-memory strategies have deterministic next-action functions. A strategy is *memoryless* if $|\mathsf{Mem}| = 1$, i.e., it does not depend on the history but only on the current state of the game.

We resp. denote by $\Lambda_i(G), \Lambda_i^F(G), \Lambda_i^{PF}(G), \Lambda_i^M(G)$ and $\Lambda_i^{PM}(G)$ the sets of general (i.e., possibly randomized and infinite-memory), finite-memory, pure finite-memory, memoryless and pure memoryless strategies for player $\mathcal{P}_i$ on the game G. We do not write G in this notation when the context is clear. A play π is said to be *consistent* with a strategy $\lambda_i \in \Lambda_i$ if for all $n \geq 0$ such that $\mathsf{Last}(\pi(n)) \in S_i$, we have $\mathsf{Last}(\pi(n+1)) \in \mathsf{Supp}(\lambda_i(\pi(n)))$.

Markov decisions processes. A finite *Markov decision process* (MDP) is a tuple $P = (\mathcal{G}, S_1, S_\Delta, \Delta)$ where (i) $\mathcal{G} = (S, E, w)$ is a finite weighted graph, (ii) S_1 and S_Δ define a partition of the set of states S into states of $\mathcal{P}_1$ and *stochastic states*, and (iii) $\Delta\colon S_\Delta \to \mathcal{D}(S)$ is the transition function that, given a stochastic state $s \in S_\Delta$, defines the probability distribution $\Delta(s)$ over the possible successors of s, such that for all states $s \in S_\Delta$, $\mathsf{Supp}(\Delta(s)) \subseteq \mathsf{Succ}(s)$.

In contrast to some other classical definitions of MDPs in the literature, we explicitly allow that, for some states $s \in S_\Delta$, $\mathsf{Supp}(\Delta(s)) \subsetneq \mathsf{Succ}(s)$: some edges of the graph $\mathcal{G}$ are assigned probability zero by the transition function. This is important as far as modeling is concerned, as in our context, transition functions will be defined according to a stochastic model for the environment of a system, and we cannot reasonably assume that such a model always involves all the possible actions of the environment. Consequently, given the MDP P, we define the subset of edges $E_\Delta = \{(s_1, s_2) \in E \mid s_1 \in S_\Delta \Rightarrow s_2 \in \mathsf{Supp}(\Delta(s_1))\}$, representing all edges that either start in a state of $\mathcal{P}_1$, or are chosen with non-zero probability by the transition function Δ.

An MDP can be seen as a two-player game where $\mathcal{P}_1$ is playing against a probabilistic adversary using a fixed randomized memoryless strategy Δ in states of the set S_Δ. Hence MDPs are sometimes referred to as $1\frac{1}{2}$-player games. The notions of prefixes belonging to $\mathcal{P}_1$ and of strategies for $\mathcal{P}_1$ are naturally extended to MDPs.

End-components. We define *end-components* (ECs) of an MDP as subgraphs in which $\mathcal{P}_1$ can ensure to stay despite stochastic states [10]. Formally, let $P = (\mathcal{G}, S_1, S_\Delta, \Delta)$ be an MDP, with $\mathcal{G} = (S, E, w)$ its underlying graph. An EC in P is a set $U \subseteq S$ such that (i) the subgraph $(U, E_\Delta \cap (U \times U))$ is strongly connected, with E_Δ defined as before, i.e., stochastic edges with probability zero are treated as non-existent; and (ii) for all $s \in U \cap S_\Delta$, $\mathsf{Supp}(\Delta(s)) \subseteq U$, i.e., in stochastic states, all outgoing edges either stay in U or belong to $E \setminus E_\Delta$ (that is, the probability of leaving U from a state $s \in S_\Delta$ is zero). The set of all ECs of P is denoted $\mathcal{E} \subseteq 2^S$.

Markov chains. A finite *Markov chain* (MC) is a tuple $M = (\mathcal{G}, \delta)$ where (i) $\mathcal{G} = (S, E, w)$ is a finite weighted graph; and (ii) $\delta \colon S \to \mathcal{D}(S)$ is the transition function that, given a state $s \in S$, defines the probability distribution $\delta(s)$ over the possible successors of s, such that for all states $s \in S$, $\mathsf{Supp}(\delta(s)) \subseteq \mathsf{Succ}(s)$.

In a Markov chain $M = (\mathcal{G}, \delta)$, an *event* is a measurable set of plays $\mathcal{A} \subseteq \mathsf{Plays}(\mathcal{G})$. Every event has a uniquely defined probability [28] (Carathéodory's extension theorem induces a unique probability measure on the Borel σ-algebra over $\mathsf{Plays}(\mathcal{G})$). We denote by $\mathbb{P}^M_{s_{\mathsf{init}}}(\mathcal{A})$ the probability that a play belongs to $\mathcal{A}$ when the Markov chain M starts in $s_{\mathsf{init}} \in S$ and is executed for an infinite number of steps. Given a measurable value function $f \colon \mathsf{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$, we denote by $\mathbb{E}^M_{s_{\mathsf{init}}}(f)$ the *expected value* or *expectation* of f over a play starting in s_{init}. The σ-algebra is defined through cylinder sets of prefixes: each prefix ρ defines a set of plays π such that ρ is a prefix of π [1]. Hence, the notions of probability and expected value can naturally be used over prefixes by considering the plays belonging to their cylinder set.

Projections. Given a set A_i, $1 \leq i \leq k$ of a cartesian product $A_1 \times \ldots \times A_k$, we define the *projection* over A_i, denoted $\mathsf{proj}_{A_i} \colon A_1 \times \ldots \times A_k \to A_i$, as the mapping from elements $\overline{a} = (a_1, \ldots, a_k)$ to $\mathsf{proj}_{A_i}(\overline{a}) = a_i$.

Outcomes. Let $M = (\mathcal{G}, \delta)$ be a Markov chain, with $\mathcal{G} = (S, E, w)$ its underlying graph. Given an initial state $s_{\mathsf{init}} \in S$, we define the set of its possible *outcomes* as

$$\mathsf{Outs}_M(s_{\mathsf{init}}) = \{\pi = s_0 s_1 s_2 \ldots \in \mathsf{Plays}(\mathcal{G}) \mid s_0 = s_{\mathsf{init}} \wedge \forall n \in \mathbb{N},\, s_{n+1} \in \mathsf{Supp}(\delta(s_n))\}\,.$$

Note that if δ is deterministic (i.e., if the support is a singleton) in all states, we obtain a unique play $\pi = s_0 s_1 s_2 \ldots$ as the unique possible outcome.

Let $G = (\mathcal{G}, S_1, S_2)$ be a two-player game, with $\mathcal{G} = (S, E, w)$ its underlying graph. Given two strategies, $\lambda_1 \in \Lambda_1$ and $\lambda_2 \in \Lambda_2$, and an initial state $s_{\mathsf{init}} \in S$, we extend the notion of outcomes as follows:

$$\mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2) = \{\pi = s_0 s_1 s_2 \ldots \in \mathsf{Plays}(\mathcal{G}) \mid s_0 = s_{\mathsf{init}} \wedge \pi \text{ is consistent with } \lambda_1 \text{ and } \lambda_2\}\,.$$

Remark that when fixing the strategies, we obtain a Markov chain denoted by $G[\lambda_1, \lambda_2]$. This MC is finite if both λ_1 and λ_2 are finite-memory strategies. Let $\mathcal{M}(\lambda_1) = (\mathsf{Mem}_1, \mathsf{m}_1, \alpha^1_{\mathsf{u}}, \alpha^1_{\mathsf{n}})$ and $\mathcal{M}(\lambda_2) = (\mathsf{Mem}_2, \mathsf{m}_2, \alpha^2_{\mathsf{u}}, \alpha^2_{\mathsf{n}})$ be the Moore machines of two such strategies. The set of states of the resulting MC is obtained through the product of the memory elements of the strategies given as Moore machines and the states of the game, i.e., $S \times \mathsf{Mem}_1 \times \mathsf{Mem}_2$; and its transition function is defined based on the distributions prescribed by the strategies and in order to accurately account for the memory updates. Notice that the outcomes of G and $G[\lambda_1, \lambda_2]$ are different objects by nature: the former are plays on a graph defined by the set of states S while the latter are plays on a graph defined by $S \times \mathsf{Mem}_1 \times \mathsf{Mem}_2$. Still, there exists a bijection between outcomes of the MC and their *traces* in the initial game, thanks to the projection operator (Lemma 1).

Lemma 1. *Let $G = (\mathcal{G}, S_1, S_2)$ be a game, with $\mathcal{G} = (S, E, w)$ its underlying graph. Let $\lambda_1 \in \Lambda_1^F$ and $\lambda_2 \in \Lambda_2^F$ be the finite-memory strategies of the players. Then there is a bijection between outcomes in G and outcomes in the resulting Markov chain $G[\lambda_1, \lambda_2]$.*

Proof. Let $s_{\mathsf{init}} \in S$ be the initial state of the game, $\mathcal{M}(\lambda_1) = (\mathsf{Mem}_1, \mathsf{m}_1, \alpha^1_{\mathsf{u}}, \alpha^1_{\mathsf{n}})$ and $\mathcal{M}(\lambda_2) = (\mathsf{Mem}_2, \mathsf{m}_2, \alpha^2_{\mathsf{u}}, \alpha^2_{\mathsf{n}})$ be the Moore machines.

Consider an outcome in $G[\lambda_1, \lambda_2]$: it is a sequence of states from $S \times \mathsf{Mem}_1 \times \mathsf{Mem}_2$. Obviously, its projection on the set S is unique and defines the outcome in the sense of G.

Conversely, consider an outcome in G: it is of the form $s_0 s_1 s_2 \ldots \in S^\omega$, with $s_0 = s_{\mathsf{init}}$. We claim there is a unique corresponding outcome in $G[\lambda_1, \lambda_2]$, written $(s_0, \mathsf{m}_1^0, \mathsf{m}_2^0)(s_1, \mathsf{m}_1^1, \mathsf{m}_2^1) \ldots \in S \times \mathsf{Mem}_1 \times \mathsf{Mem}_2$, with $(s_0, \mathsf{m}_1^0, \mathsf{m}_2^0) = (s_{\mathsf{init}}, \mathsf{m}_1, \mathsf{m}_2)$. Indeed, it suffices to see that the update functions of the Moore machines, α_{u}^1 and α_{u}^2, are deterministic functions. Hence, it is easy to reconstruct the outcome of $G[\lambda_1, \lambda_2]$ based on its projection on S as it suffices to apply the effect of the update functions on the memory at each step. □

Hence, we obtain the following equality:

$$\mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2) = \mathsf{proj}_S\left(\mathsf{Outs}_{G[\lambda_1,\lambda_2]}((s_{\mathsf{init}}, \mathsf{m}_1, \mathsf{m}_2))\right).$$

Based on this, and for the sake of readability, we abuse the notation and write $\mathsf{Outs}_{G[\lambda_1,\lambda_2]}(s_{\mathsf{init}})$ equivalently to refer to this set of outcomes. Similar abuse is taken for value functions and initial states.

Back to the outcomes of the game: note that if both strategies λ_1 and λ_2 are pure, the resulting Markov chain only involves Dirac distributions (δ is deterministic) and the set $\mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2)$ is composed of a unique play $\pi = s_0 s_1 s_2 \ldots$ such that for all $n \geq 0$, $i \in \{1,2\}$, if $s_n \in S_i$, then we have $\lambda_i(s_n) = s_{n+1}$.

Let $P = (\mathcal{G}, S_1, S_\Delta, \Delta)$ be a Markov decision process, with $\mathcal{G} = (S, E, w)$ its underlying graph. Again, we can fix the strategy λ_1 of $\mathcal{P}_1$ and obtain the Markov chain $P[\lambda_1]$. Let $\mathcal{M}(\lambda_1) = (\mathsf{Mem}, \mathsf{m}_0, \alpha_{\mathsf{u}}, \alpha_{\mathsf{n}})$. The set of outcomes starting in $s_{\mathsf{init}} \in S$ is defined as $\mathsf{Outs}_P(s_{\mathsf{init}}, \lambda_1) = \mathsf{proj}_S\left(\mathsf{Outs}_{P[\lambda_1]}((s_{\mathsf{init}}, \mathsf{m}_0))\right)$. Again, we abuse the notation and write $\mathsf{Outs}_{P[\lambda_1]}(s_{\mathsf{init}})$ equivalently.

Finally, back to the two-player game G, if we fix the strategy λ_i of only one player $\mathcal{P}_i$, $i \in \{1,2\}$, we obtain not a Markov chain, but a Markov decision process for the remaining player $\mathcal{P}_{3-i}$. This MDP is denoted by $G[\lambda_i]$. We define its set of outcomes as

$$\mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_i) = \bigcup_{\lambda_{3-i} \in \Lambda_{3-i}} \mathsf{Outs}_{G[\lambda_i]}(s_{\mathsf{init}}, \lambda_{3-i}) = \bigcup_{\lambda_{3-i} \in \Lambda_{3-i}} \mathsf{Outs}_{G[\lambda_1,\lambda_2]}(s_{\mathsf{init}}).$$

Attractors. Given a game $G = (\mathcal{G}, S_1, S_2)$, with $\mathcal{G} = (S, E, w)$, the *attractor* for $\mathcal{P}_1$ of a set $A \subseteq S$ in G is denoted by $\mathsf{Attr}_G^{\mathcal{P}_1}(A)$ and computed as the fixed point of the sequence

$$\mathsf{Attr}_G^{\mathcal{P}_1,n+1}(A) = \mathsf{Attr}_G^{\mathcal{P}_1,n}(A) \cup \{s \in S_1 \mid \exists (s,t) \in E, t \in \mathsf{Attr}_G^{\mathcal{P}_1,n}(A)\} \cup \{s \in S_2 \mid \forall (s,t) \in E, t \in \mathsf{Attr}_G^{\mathcal{P}_1,n}(A)\},$$

with $\mathsf{Attr}_G^{\mathcal{P}_1,0}(A) = A$. The attractor $\mathsf{Attr}_G^{\mathcal{P}_1}(A)$ is exactly the set of states from which $\mathcal{P}_1$ can ensure to reach A no matter what $\mathcal{P}_2$ does. That is,

$$\mathsf{Attr}_G^{\mathcal{P}_1}(A) = \{s \in S \mid \exists \lambda_1 \in \Lambda_1(G), \forall \lambda_2 \in \Lambda_2(G), \forall \pi = s_0 s_1 s_2 \ldots \in \mathsf{Outs}_G(s, \lambda_1, \lambda_2), s_0 = s, \exists i \in \mathbb{N}, s_i \in A\}$$

The attractor $\mathsf{Attr}_G^{\mathcal{P}_2}(A)$ for $\mathcal{P}_2$ is defined symmetrically.

Subgraphs, subgames and sub-MDPs. Given a graph $\mathcal{G} = (S, E, w)$ and a subset of states $A \subseteq S$, we define the induced subgraph $\mathcal{G} \restriction A = (A, E \cap (A \times A), w)$ naturally. Subgames and sub-MDPs are defined similarly by considering their induced subgraphs. It is to note that subgames and sub-MDPs can only be properly defined if the induced subgraphs contain no deadlock and if the transition functions remain well-defined in the case of MDPs (i.e., if the probabilities on outgoing edges still sum up to one in all stochastic states of the sub-MDP).

Worst-case synthesis. Given a two-player game $G = (\mathcal{G}, S_1, S_2)$, with $\mathcal{G} = (S, E, w)$ its underlying graph, an initial state $s_{\mathsf{init}} \in S$, a value function $f \colon \mathsf{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$, and a rational threshold $\mu \in \mathbb{Q}$, the *worst-case threshold problem* asks to decide if $\mathcal{P}_1$ has a strategy $\lambda_1 \in \Lambda_1$ such that

$$\forall \lambda_2 \in \Lambda_2, \forall \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2),\ f(\pi) \geq \mu.$$

For the mean-payoff value function, pure memoryless optimal[5] strategies exist for both players [24,13]. Hence, deciding the winner is in NP ∩ coNP, and it was furthermore shown to be in UP ∩ coUP [30,23,17]. Whether the

[5] A strategy for $\mathcal{P}_i$, $i \in \{1,2\}$, is said to be *optimal* if it ensures a threshold higher or equal to the threshold ensured by any other strategy of the same player. The threshold ensured by an optimal strategy is called the *optimal value*.

problem is in P is a long-standing open problem [3,6]. Total-payoff value functions also yield pure memoryless optimal strategies for both players [18] and the associated decision problem is in UP $\cap$ coUP [17]. For the truncated sum function, which can be seen as a particular instance of total-payoff, it can be shown that the decision problem takes polynomial time, as a winning strategy of $\mathcal{P}_1$ should avoid all cycles (because they yield strictly positive costs), hence usage of attractors and comparison of the worst possible sum of costs with the threshold suffices.

Expected value synthesis. Given an MDP $P = (\mathcal{G}, S_1, S_\Delta, \Delta)$, with $\mathcal{G} = (S, E, w)$ its underlying graph, an initial state $s_{\mathsf{init}} \in S$, a measurable value function $f\colon \mathsf{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$, and a rational threshold $\nu \in \mathbb{Q}$, the *expected value threshold problem* asks to decide if $\mathcal{P}_1$ has a strategy $\lambda_1 \in \Lambda_1$ such that

$$\mathbb{E}_{s_{\mathsf{init}}}^{P[\lambda_1]}(f) \geq \nu.$$

Optimal expected mean-payoff in MDPs can be achieved by memoryless strategies, and the corresponding decision problem can be solved in polynomial time through linear programming [14]. The truncated sum value function has been studied in the literature under the name of *shortest path problem*: again, memoryless strategies suffice to be optimal and the problem is solvable in polynomial time via linear programming [2,11].

3 Beyond Worst-Case Synthesis

We here define the *beyond worst-case synthesis problem*. Our goal is to study the synthesis of finite-memory strategies that, *simultaneously*, ensure a value greater than some threshold μ in the worst-case situation (i.e., against any strategy of the adversary), and ensure an expected value greater than some threshold ν against a given finite-memory stochastic model of the adversary (e.g., representing commonly observed behavior of the environment).

Definition 1. *Given a game $G = (\mathcal{G}, S_1, S_2)$, with $\mathcal{G} = (S, E, w)$ its underlying graph, an initial state $s_{\mathsf{init}} \in S$, a finite-memory stochastic model $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^F$ of the adversary, represented by a stochastic Moore machine, a measurable value function $f\colon \mathsf{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$, and two rational thresholds $\mu, \nu \in \mathbb{Q}$, the* beyond worst-case (BWC) problem *asks to decide if $\mathcal{P}_1$ has a finite-memory strategy $\lambda_1 \in \Lambda_1^F$ such that*

$$\begin{cases} \forall\, \lambda_2 \in \Lambda_2, \forall\, \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2),\ f(\pi) > \mu & (1) \\ \mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1, \lambda_2^{\mathsf{stoch}}]}(f) > \nu & (2) \end{cases}$$

and the BWC synthesis problem asks to synthesize such a strategy if one exists.

We take the convention to ask for values strictly greater than the thresholds in order to ease the formulation of our results in the following. Indeed, we will show that for some thresholds, it is possible to synthesize strategies that ensure ε-close values, for any $\varepsilon > 0$, while it is not feasible to achieve the exact threshold (Sect. 4.11). Using the strict inequality, we avoid tedious manipulation of such ε in our proofs. Notice that we can assume $\nu > \mu$, otherwise the problem reduces to the classical worst-case analysis as follows. Assume $\mu \geq \nu$ and $\lambda_1^{pm} \in \Lambda_1^{PM}$ satisfies the worst-case threshold (recall memory is not necessary for the worst-case requirement alone). Consider the MC $G[\lambda_1^{pm}, \lambda_2^{\mathsf{stoch}}]$. By eq. (1) and Lemma 1, we have that for all $\pi \in \mathsf{Outs}_{G[\lambda_1^{pm}, \lambda_2^{\mathsf{stoch}}]}(s_{\mathsf{init}})$, $f(\pi) > \mu$. Hence, regardless of how the probability is defined in the MC, we have that $\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^{pm}, \lambda_2^{\mathsf{stoch}}]}(f) > \mu \geq \nu$ and eq. (2) is trivially satisfied.

4 Mean-Payoff Value Function

The first value function that we study is the mean-payoff. We present an algorithm, BWC_MP (Alg. 1), for deciding the corresponding BWC problem. Its cornerstones are highlighted in Sect. 4.1. A running example is presented in Sect. 4.2. Sections 4.3 through 4.8 are devoted to a detailed justification of this algorithm and the proof of its correctness. In Sect. 4.9, we prove that the complexity of algorithm BWC_MP is NP$\cap$coNP and that it is optimal with regard to the complexity of the worst-case problem. Thus, the BWC framework for mean-payoff surprisingly provides additionnal modeling power without negative impact on the complexity class. In Sect. 4.10, we prove that pseudo-polynomial memory is sufficient and in general necessary for finite-memory strategies satisfying the BWC mean-payoff problem (polynomial in the size of the game and the stochastic model, and in the values of weights and thresholds). Finally, we show in Sect. 4.11 that infinite-memory strategies are strictly more powerful than finite-memory ones for $\mathcal{P}_1$. This is in contrast with the worst-case and the expected value settings, where memoryless strategies always suffice.

Algorithm 1 BWC_MP$(G^i, \lambda_2^i, \mu^i, \nu^i, s_{\mathsf{init}}^i)$

Require: $G^i = \left(\mathcal{G}^i, S_1^i, S_2^i\right)$ a game, $\mathcal{G}^i = \left(S^i, E^i, w^i\right)$ its underlying graph, $\lambda_2^i \in \Lambda_2^F(G^i)$ a finite-memory stochastic model of the adversary, $\mathcal{M}(\lambda_2^i) = (\mathsf{Mem}, \mathsf{m}_0, \alpha_\mathsf{u}, \alpha_\mathsf{n})$ its Moore machine, $\mu^i = \frac{a}{b}, \nu^i \in \mathbb{Q}$, $\mu^i < \nu^i$, resp. the worst-case and the expected value thresholds, and $s_{\mathsf{init}}^i \in S^i$ the initial state

Ensure: The answer is YES if and only if $\mathcal{P}_1$ has a finite-memory strategy $\lambda_1 \in \Lambda_1^F(G^i)$ satisfying the BWC problem from s_{init}^i, for the thresholds pair (μ^i, ν^i) and the mean-payoff value function

$\{$*Preprocessing*$\}$

1: **if** $\mu^i \neq 0$ **then**
2: Modify the weight function of $\mathcal{G}^i$ s.t. $\forall e \in E^i$, $w_{\mathsf{new}}^i(e) := b \cdot w^i(e) - a$, and consider the new thresholds pair $(0, \nu := b \cdot \nu^i - a)$
3: Compute $S_{WC} := \left\{s \in S^i \mid \exists \lambda_1 \in \Lambda_1(G^i), \forall \lambda_2 \in \Lambda_2(G^i), \forall \pi \in \mathsf{Outs}_{G^i}(s, \lambda_1, \lambda_2), \mathsf{MP}(\pi) > 0\right\}$
4: **if** $s_{\mathsf{init}}^i \notin S_{WC}$ **then**
5: **return** NO
6: **else**
7: Let $G^w := G^i \restriction S_{WC}$ be the subgame induced by worst-case winning states
8: Build $G := G^w \otimes \mathcal{M}(\lambda_2^i) = (\mathcal{G}, S_1, S_2)$, $\mathcal{G} = (S, E, w)$, $S \subseteq (S_{WC} \times \mathsf{Mem})$, the game obtained by product with the Moore machine, and $s_{\mathsf{init}} := (s_{\mathsf{init}}^i, \mathsf{m}_0)$ the corresponding initial state
9: Let $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^M(G)$ be the memoryless transcription of λ_2^i on G
10: Let $P := G[\lambda_2^{\mathsf{stoch}}] = (\mathcal{G}, S_1, S_\Delta = S_2, \Delta = \lambda_2^{\mathsf{stoch}})$ be the MDP obtained from G and $\lambda_2^{\mathsf{stoch}}$

$\{$*Main algorithm*$\}$

11: Compute $\mathcal{U}_\mathsf{w}$ the set of maximal winning end-components of P
12: Build $P' = (\mathcal{G}', S_1, S_\Delta, \Delta)$, where $\mathcal{G}' = (S, E, w')$ and w' is defined as follows:

$$\forall e = (s_1, s_2) \in E,\ w'(e) := \begin{cases} w(e) \text{ if } \exists U \in \mathcal{U}_\mathsf{w} \text{ s.t. } \{s_1, s_2\} \subseteq U \\ 0 \text{ otherwise} \end{cases}$$

13: Compute the maximal expected value ν^* from s_{init} in P'
14: **if** $\nu^* > \nu$ **then**
15: **return** YES
16: **else**
17: **return** NO

4.1 The approach in a nutshell

Algorithm BWC_MP is described in Alg. 1. We give an intuitive sketch of its functioning in the following.

Inputs and outputs. The algorithm takes as input: a game G^i, a finite-memory stochastic model of the adversary λ_2^i, a worst-case threshold μ^i, an expected value threshold ν^i, and an initial state s_{init}^i. Its output is YES if and only if there exists a finite-memory strategy of $\mathcal{P}_1$ satisfying the BWC problem (Def. 1).

The output as described in Alg. 1 is boolean: the algorithm answers whether a satisfying strategy exists or not, but does not explicitely construct it (to avoid tedious formalization within the pseudocode). Nevertheless, we present how to synthesize such a winning strategy in Sect. 4.8. We sketch its operation in the following and we highlight the role of each step of the algorithm in the construction of this winning strategy, as producing a witness winning strategy is a straightforward by-product of the process we apply to decide satisfaction of the BWC problem.

Preprocessing. The first part of the algorithm (lines 1 through 10) is dedicated to the preprocessing of the game G^i and the stochatic model λ_2^i given as inputs in order to apply the second part of the algorithm (lines 11 through 17) on a modified game G and stochastic model $\lambda_2^{\mathsf{stoch}}$, simpler to manipulate. We show in the following that the answer to the BWC problem on the modified game is YES if and only if it is also YES on the input game, and we present how a winning strategy of $\mathcal{P}_1$ in G can be transferred to a winning strategy in G^i.

The preprocessing is composed of four main steps. First, we modify the weight function of $\mathcal{G}^i$ in order to consider the equivalent BWC problem with thresholds $(0, \nu)$ instead of (μ^i, ν^i). This classical trick is used to get rid of explicitely considering the worst-case threshold in the following, as it is equal to zero. Second, remark that any strategy that is winning for the BWC problem must also be winning for the classical worst-case problem. Such a strategy cannot

allow visits of any state from which $\mathcal{P}_1$ cannot ensure winning against an antagonistic adversary because mean-payoff is a prefix-independent[6] objective (hence it is not possible to "win" it over the finite prefix up to such a state). Hence, we reduce our study to G^w, the subgame induced by worst-case winning states in G^i (lines 3 and 7). Obviously, if from the initial state s^i_{init}, $\mathcal{P}_1$ cannot win the worst-case problem, then the answer to the BWC problem is NO (lines 4-5). Third, we build the game G which states are defined by the product of the states of G^w and the memory elements of the Moore machine $\mathcal{M}(\lambda_2^i)$ (line 8). Intuitively, we expand the initial game by integrating the memory of the stochastic model of $\mathcal{P}_2$ in the graph. Note that this does not modify the power of the adversary. Fourth, the finite-memory stochastic model λ_2^i on G^i clearly translates to a memoryless stochastic model $\lambda_2^{\mathsf{stoch}}$ on G (line 9). This will help us obtain elegant proofs for the second part of the algorithm.

Analysis of end-components. The second part of the algorithm (lines 11-17) hence operates on a game G such that from all states, $\mathcal{P}_1$ has a strategy to achieve a strictly positive mean-payoff value (recall $\mu = 0$). We consider the MDP $P = G[\lambda_2^{\mathsf{stoch}}]$ and notice that the underlying graphs of G and P are the same thanks to $\lambda_2^{\mathsf{stoch}}$ being memoryless. The following steps rely on the analysis of *end-components* in the MDP, i.e., strongly connected subgraphs in which $\mathcal{P}_1$ can ensure to stay when playing against the stochastic adversary (cf. Sect. 2).

The motivation to the analysis of ECs is the following. It is well-known that under any arbitrary strategy $\lambda_1 \in \Lambda_1$ of $\mathcal{P}_1$ in P, the probability that states visited infinitely often along an outcome constitute an EC is one [9,10]. Recall that the mean-payoff is prefix-independent, therefore the value of any outcome only depends on those states that are seen infinitely often. Hence, the expected mean-payoff in $P[\lambda_1]$ depends *uniquely* on the value obtained in the ECs. Inside an EC, we can compute the maximal expected value that can be achieved by $\mathcal{P}_1$, and this value is the same in all states of the EC [14].

Consequently, in order to satisfy the expected value requirement (eq. (2)), an acceptable strategy for the BWC problem has to favor reaching ECs with a sufficient expectation, but under the constraint that it should also ensure satisfaction of the worst-case requirement (eq. (1)). As we will show in the following, this constraint implies that some ECs with high expected values may still need to be avoided because they do not permit to guarantee the worst-case requirement. This is the cornerstone of the classification of ECs that follows.

Classification of end-components. Let $\mathcal{E} \subseteq 2^S$ be the set of all ECs in P. Notice that by definition, only edges in E_Δ, as defined in Sect. 2, are involved to determine which sets of states form an EC in P. As such, for any EC $U \in \mathcal{E}$, there may exist edges from $E \setminus E_\Delta$ starting in U, such that $\mathcal{P}_2$ can force leaving U when using an arbitrary strategy. Still these edges will never be used by the stochastic model $\lambda_2^{\mathsf{stoch}}$. This remark will be important to the definition of strategies of $\mathcal{P}_1$ that guarantee the worst-case requirement, as $\mathcal{P}_1$ needs to be able to react to the hypothetic use of such an edge. We will see that it is also the case *inside* an EC.

Now, we want to consider the ECs in which $\mathcal{P}_1$ can ensure that the worst-case requirement will be fulfilled (i.e., without having to leave the EC): we call them *winning* ECs. Indeed, the others will need to be eventually avoided, hence will have zero impact on the expectation of a finite-memory strategy satisfying the BWC problem. So we call the latter *losing* ECs. The subtlety of this classication is that it involves considering the ECs both in the MDP P, and in the game G. Formally, let $U \in \mathcal{E}$ be an EC. It is *winning* if, in the subgame $G \restriction U$, from all states, $\mathcal{P}_1$ has a strategy to ensure a strictly positive mean-payoff against any strategy of $\mathcal{P}_2$ *that only chooses edges which are assigned non-zero probability by* $\lambda_2^{\mathsf{stoch}}$, or equivalently, edges in E_Δ. We denote $\mathcal{W} \subseteq \mathcal{E}$ the set of such ECs. Non-winning ECs are *losing*: in those, whatever the strategy of $\mathcal{P}_1$ played against the stochastic model $\lambda_2^{\mathsf{stoch}}$ (or any strategy with the same support), there exists at least one outcome for which the mean-payoff is not strictly positive (even if its probability is zero, its mere existence is not acceptable for the worst-case requirement).

Maximal winning end-components. Based on these definitions, remark that line 11 of algorithm BWC_MP does not actually compute the set $\mathcal{W}$ containing all winning ECs, but rather the set $\mathcal{U}_{\mathrm{w}} \subseteq \mathcal{W}$, defined as $\mathcal{U}_{\mathrm{w}} = \{U \in \mathcal{W} \mid \forall U' \in \mathcal{W}, U \subseteq U' \Rightarrow U = U'\}$, i.e., the set of *maximal* winning ECs.

The intuition on *why we can* restrict our study to this subset is as follows. If an EC $U_1 \in \mathcal{W}$ is included in another EC $U_2 \in \mathcal{W}$, i.e., $U_1 \subseteq U_2$, we have that the maximal expected value achievable in U_2 is at least equal to the one achievable in U_1. Indeed, by definition, U_2 is strongly connected and $\mathcal{P}_1$ can force reaching U_1 and staying in it forever with probability one (by virtue of U_1 being an EC): hence the expectation of such a strategy would be equal to what

[6] For all $\rho \in \mathsf{Prefs}(\mathcal{G})$, $\pi \in \mathsf{Plays}(\mathcal{G})$ we have that $\mathsf{MP}(\rho \cdot \pi) = \mathsf{MP}(\pi)$.

can be obtained in U_1 thanks to the prefix-independence of the mean-payoff. This property implies that it is sufficient to consider maximal winning ECs in our computations.

As for *why we do it*, observe that the complexity gain is critical. The number of winning ECs can be as large as $|\mathcal{W}| \leq |\mathcal{E}| \leq 2^{|S|}$, that is, exponential in the size of the input. Yet, the number of maximal winning ECs is bounded by $|\mathcal{U}_\mathrm{w}| \leq |S|$ as they are disjoint by definition. Indeed, for any two winning ECs with a non-empty intersection, their union also constitutes an EC, and is still winning because $\mathcal{P}_1$ can essentially stick to the EC of his choice. The computation of the set $\mathcal{U}_\mathrm{w}$ is executed by a recursive subalgorithm which is in NP $\cap$ coNP. Roughly sketched, this algorithm computes the maximal end-component decomposition of an MDP (in polynomial time [7]), then checks for each EC U in the decomposition (their number is polynomial) if U is winning or not, which requires a call to an NP $\cap$ coNP oracle solving the worst-case threshold problem on the corresponding subgame. If U is losing, it may still be the case that a sub-EC $U' \subsetneq U$ is winning. Therefore we recurse on the MDP reduced to U, where states from which $\mathcal{P}_2$ can win in U have been removed (they are a no-go for $\mathcal{P}_1$). Hence the stack of calls is also at most polynomial.

Ensure reaching winning end-components. As discussed, under any arbitrary strategy of $\mathcal{P}_1$, states visited infinitely often form an EC with probability one. Now, if we take a *finite-memory* strategy that *satisfies* the BWC problem (Def. 1), we can precise this result and state that they form a *winning* EC with probability one. Equivalently, we have that the probability that an outcome π is such that $\mathsf{Inf}(\pi) = U$ for some $U \in \mathcal{E} \setminus \mathcal{W}$ is zero. Remark the equality is crucial. It may be the case, with non-zero probability, that $\mathsf{Inf}(\pi) = U' \subsetneq U$ for some $U' \in \mathcal{W}$ and $U \in \mathcal{E} \setminus \mathcal{W}$ (hence the recursive algorithm to compute $\mathcal{U}_\mathrm{w}$). It is clear that $\mathcal{P}_1$ should not visit all the states of a losing EC forever, as then he would not be able to guarantee the worst-case threshold inside the corresponding subgame.[7]

We denote $S_{\mathsf{neg}} = S \setminus \bigcup_{U \in \mathcal{U}_\mathrm{w}} U$ the set of states that, with probability one, are only seen a finite number of times when a BWC satisfying strategy is played, and call them *negligible* states.

Our ultimate goal here is to build a modified MDP P', sharing the same underlying graph and ECs as P, such that a classical optimal strategy for the expected value problem on P' will naturally avoid losing ECs and prescribe which winning ECs are the most interesting to reach for a BWC strategy on the initial game G and MDP P. Observe that the expected value obtained in P by any BWC satisfying strategy of $\mathcal{P}_1$ only depends on the weights of edges involved in winning ECs, or equivalently, in maximal winning ECs (as the set of outcomes that are not trapped in them has measure zero). Consequently, we build P' by modifying the weight function of P (line 12). Basically, we keep the weights unchanged in edges that belong to some $U \in \mathcal{U}_\mathrm{w}$, and we put them to zero everywhere else, i.e., on any edge involving a negligible state. Weight zero is taken because it is lower than the expectation granted by winning ECs, which is strictly greater than zero by definition.

Reach the highest valued winning end-components. We compute the maximal expected mean-payoff v^* that can be achieved by $\mathcal{P}_1$ in the MDP P', from the corresponding initial state (line 13). This computation takes polynomial time and memoryless strategies suffice to achieve the maximal value [14].

As discussed before, such a strategy reaches an EC of P' with probability one. Basically, we build a strategy that favors reaching ECs with high associated expectations in P'. We argue that the ECs reached with probability one by this strategy are necessarily winning ECs. Clearly, if a winning EC is reachable instead of a losing one, it will be favored because of the weights definition in P' (expectation is strictly higher in winning ECs). Thus it remains to check if the set of winning ECs is reachable with probability one from any state in S. That is the case because of the preprocessing. Indeed, we know that all states are winning for the worst-case requirement. Clearly, from any state in $A = S \setminus \bigcup_{U \in \mathcal{E}} U$, $\mathcal{P}_1$ cannot ensure to stay in A (otherwise it would form an EC) and thus must be able to win the worst-case requirement from reached ECs. Now for any state in $B = \bigcup_{U \in \mathcal{E}} U \setminus \bigcup_{U \in \mathcal{U}_\mathrm{w}} U$, i.e., states in losing ECs and not in any sub-EC winning, $\mathcal{P}_1$ cannot win the worst-case by staying in B, by definition of losing EC. Since we know $\mathcal{P}_1$ can ensure the worst-case by hypothesis, it is clear that he must be able to reach $C = \bigcup_{U \in \mathcal{U}_\mathrm{w}} U$ from any state in B, as claimed.

Inside winning end-components. Based on that, winning ECs are reached with probability one. Let us first consider what we can say about such ECs if we assume that $E_\Delta = E$, i.e., if the stochastic model maps all possible edges to non-zero probabilities. We establish a finite-memory *combined strategy* of $\mathcal{P}_1$ that ensures (i) worst-case satisfaction while yielding (ii) an expected value ε-close to the maximal expectation inside the component. For two well-chosen parameters $K, L \in \mathbb{N}$, it is informally defined as follows: in phase *(a)*, play a memoryless expected value optimal

[7] We show in Sect. 4.11 that with infinite memory, there may still be some incentive to stay in a losing EC.

strategy for K steps and memorize $\mathsf{Sum} \in \mathbb{Z}$, the sum of weights along these steps; in phase *(b)*, if $\mathsf{Sum} > 0$, go to *(a)*, otherwise play a memoryless worst-case optimal strategy for L steps, then go to *(a)*. In phases *(a)*, $\mathcal{P}_1$ tries to increase its expectation and approach its optimal one, while in phase *(b)*, he compensates, if needed, losses that occured in phase *(a)*. The two memoryless strategies exist on the subgame induced by the EC: by definition of ECs, based on E_Δ, the stochastic model of $\mathcal{P}_2$ will never be able to force leaving the EC against the combined strategy. A key result of our paper is the existence of values for K and L such that (i) and (ii) are verified. We see plays as sequences of periods, each starting with phase *(a)*. First, for any K, it is possible to define $L(K)$ such that any period composed of phases *(a)* + *(b)* ensures a mean-payoff at least $1/(K+L) > 0$. Periods containing only phase *(a)* trivially induce a mean-payoff at least $1/K$ as they are not followed by phase *(b)*. Both rely on the weights being integers. As the length of any period is bounded by $(K+L)$, the inequality remains strict for the mean-payoff of any play, granting (i). Now, consider parameter K. Clearly, when $K \to \infty$, the expectation over a phase *(a)* tends to the optimal one. Nevertheless, phases *(b)* also contribute to the overall expectation of the combined strategy, and (in general) lower it so that it is strictly less than the optimal for any $K, L \in \mathbb{N}$. Hence to prove (ii), we not only need that the probability of playing phase *(b)* decreases when K increases, but also that it decreases faster than the increase of L, needed to ensure (i), so that overall, the contribution of phases *(b)* tends to zero when $K \to \infty$. This is indeed the case and is proved using results bounding the probability of observing a mean-payoff significantly (more than some ε) different than the optimal expectation along a phase *(a)* of length $K \in \mathbb{N}$: this probability decreases exponentially when K increases [27,19] (related to the notions of Chernoff bounds and Hoeffding's inequality in MCs), while L only needs to be polynomial in K.

Now, consider what happens if $E_\Delta \subsetneq E$. Then, if $\mathcal{P}_2$ uses an arbitrary strategy, he can take edges of probability zero, i.e., in $E \setminus E_\Delta$, either staying in the EC, or leaving it. In both cases, this must be taken into account in order to satisfy eq. (1) as it may involve dangerous weights (recall that zero-probability edges are not considered when an EC is classified as winning or not). Fortunately, if this were to occur, $\mathcal{P}_1$ could switch to a worst-case winning memoryless strategy, which exists in all states thanks to the preprocessing, to preserve the worst-case requirement. Regarding the expected value (eq. (2)), this has no impact as it occurs with probability zero against $\lambda_2^{\mathsf{stoch}}$. The strategy to follow in winning ECs hence adds this reaction procedure to the combined strategy: we call it the *witness-and-secure strategy*.

Global strategy synthesis. In summary, (i) losing ECs should be avoided and will be by a strategy that optimizes the expectation on the MDP P'; (ii) in winning ECs, $\mathcal{P}_1$ can obtain the expectation of the EC (at some arbitrarily low ε close) *and* ensure the worst-case threshold.

Hence, we finally compare the value v^* with the expected value threshold v (line 14): (i) if it is strictly higher, we conclude that there exists a finite-memory strategy satisfying the BWC problem, and (ii) if it is not, we conclude that there does not exist such a strategy. To prove (i), we establish a finite-memory strategy in G, called *global strategy*, of $\mathcal{P}_1$ that ensures a strictly positive mean-payoff against an antagonistic adversary, and ensures an expected mean-payoff ε-close to v^* (hence, strictly greater than v) against the stochastic adversary modeled by $\lambda_2^{\mathsf{stoch}}$ (i.e., in P). The intuition is as follows. We play the memoryless optimal strategy of the MDP P' for a sufficiently long time, defined by a parameter $N \in \mathbb{N}$, in order to be with probability close to one in a winning EC (the convergence is exponential by results on absorption times in MCs [20]). Then, if we are inside a winning EC, we switch to the witness-and-secure strategy which, as sketched in the previous paragraph, ensures the worst-case and the expectation thresholds. If we are not yet in a winning EC, then we switch to a worst-case winning strategy in G, which always exists by hypothesis. Thus the mean-payoff of plays that do not reach winning ECs is strictly positive. Since in winning ECs we are ε-close to the maximal expected value of the EC, we can conclude that it is possible to play the optimal expectation strategy of MDP P' for sufficiently long to obtain an overall expected value which is arbitrarily close to v^*, and still guarantee the worst-case threshold in all outcomes. To prove (ii), it suffices to understand that only ECs have an impact on the expectation, and that losing ECs cannot be used forever without endangering the worst-case requirement. Note that given a winning strategy on G, it is possible to build a corresponding winning strategy on G^i by reintegrating the memory elements of the Moore machine in the memory of the winning strategy of $\mathcal{P}_1$.

Complexity bounds. The input size of the algorithm depends on the size of the game, the size of the Moore machine for the stochastic model, and the encodings of weights and thresholds. We can prove that all computing steps require (deterministic) polynomial time except for calls to an algorithm solving the worst-case threshold problem, which is in NP $\cap$ coNP [30,23] and not known to be in P. Hence, the overall complexity of the algorithm is in NP $\cap$ coNP and may collapse to P if the worst-case problem were to be proved in P.

We also establish that the BWC problem is at least as difficult as the worst-case problem thanks to a polynomial time reduction from the latter to the former. Thus, BWC_MP membership to NP ∩ coNP can be seen as optimal regarding our current knowledge of the worst-case threshold problem.

Theorem 1. *The beyond worst-case problem for the mean-payoff value function is in NP ∩ coNP and at least as hard as deciding the winner in mean-payoff games.*

Remark 1 (approximation of the optimal value). Given a worst-case threshold $\mu \in \mathbb{Q}$, a natural question is whether we can maximize the expectation of finite-memory strategies that satisfy this threshold. However, there is no best expectation value in general, as increasing the size of the memory may also strictly increase the expectation. Nevertheless, the least upper bound of all the expected value thresholds that can be achieved by finite-memory strategies can be approached up to an ε, for all $\varepsilon > 0$. Formally, assume that the worst-case threshold μ can be satisfied, and let $\nu_\top$ be the least upper bound of the set $\{\nu \in \mathbb{Q} \mid \exists \lambda_1 \in \Lambda_1^{PF}$ that satisfies the BWC problem for thresholds $(\mu, \nu)\}$ (it exists since this set is trivially bounded by W and it is non-empty). Without knowing $\nu_\top$ a priori, it is still possible to approach it ε-closely, for all $\varepsilon > 0$, by a dichotomic search with a polynomial (in $V = \log_2 W$, the length of the encoding of weights) number of steps, initialized to the interval $[\mu, W]$.

4.2 Running example

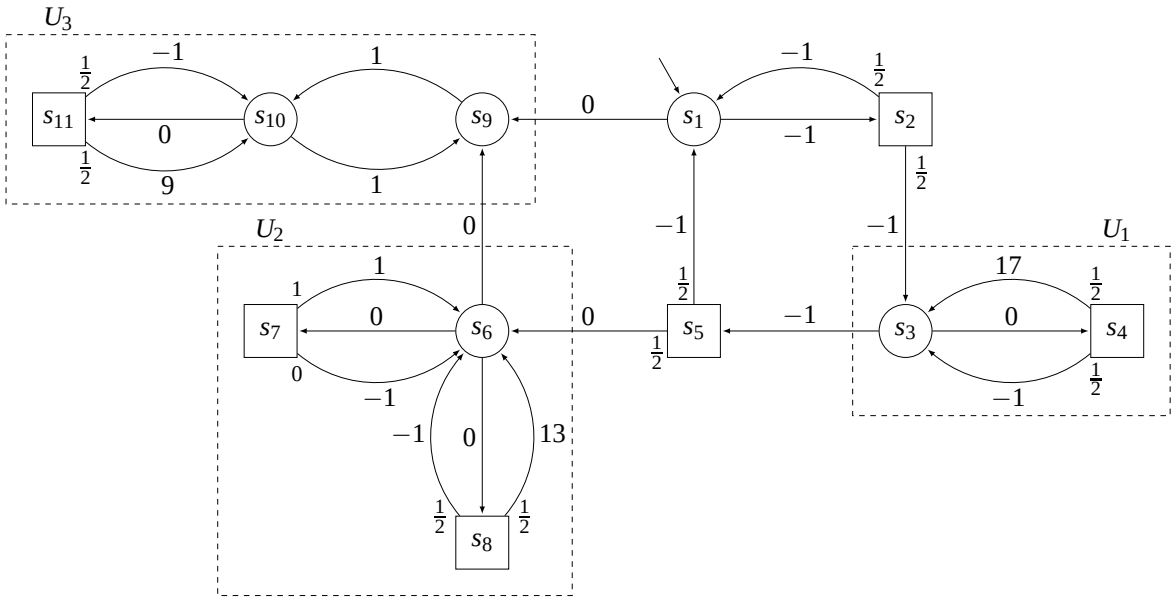

Fig. 2: Mean-payoff game with maximal winning ECs U_2 and U_3. End-component U_1 is losing.

In order to illustrate several notions and strategies, we will consider the game depicted in Fig. 2 throughout Sect. 4. States of $\mathcal{P}_1$ are represented by circles and states of $\mathcal{P}_2$ by squares. The stochastic model of $\mathcal{P}_2$ is memoryless and is described by the probabilities written close to the start of outgoing edges. For example, in s_2, the stochastic model chooses edge (s_2, s_1) with probability $1/2$ and edge (s_2, s_3) with probability $1/2$. Each edge is assigned a weight represented by an integer number midway. Formally, our definition of the set E allows only one edge from any given state s to any state s', hence asks that multiple edges with different values be split by adding dummy states (states with exactly one ingoing edge and one outgoing edge). Note that in order to preserve the same mean-payoff values for paths in the graph, we need to split every edge and copy its weight in both halfs. This restriction is w.l.o.g. and applied for the sake of readability in technical proofs. Still, our graphical representation oversteps it to maintain compactness.

We consider the BWC problem with the worst-case threshold $\mu = 0$. Remark that this game satisfies the assumptions guaranteed at the end of the preprocessing part of the algorithm. That is, the worst-case threshold is zero, a worst-case winning strategy of $\mathcal{P}_1$ exists in all states (e.g., the memoryless strategy choosing edges (s_1, s_9), (s_3, s_5), (s_6, s_9), (s_9, s_{10}) and (s_{10}, s_9) in their respective starting states), and the stochastic model is memoryless, as explained above.

4.3 Preprocessing - simplifying assumptions

We discuss here the preprocessing part of the algorithm (lines 1-10). The goal is to be able to execute the main algorithm (lines 11-17) with the following hypotheses: (a) the worst-case threshold is zero, (b) in all states, $\mathcal{P}_1$ has a strategy to satisfy the worst-case requirement, and (c) the stochastic model of the adversary is memoryless. This preprocessing is sound and complete: $\mathcal{P}_1$ has a strategy for the BWC problem in the input G^i (for thresholds (μ^i, ν^i) and against stochastic model λ_2^i) if and only if he also has one in the preprocessed game G (for thresholds $(0, \nu)$ and against stochastic model $\lambda_2^{\mathsf{stoch}}$). We break down the proof in three lemmas, one for each hypothesis.

Thresholds. First, Lemma 2 states that the worst-case threshold can be taken equal to zero thanks to a slight modification of the weight function (lines 1-2). From now on, we thus assume that $\mu = 0$.

Lemma 2. *Let $G = (\mathcal{G}, S_1, S_2)$ be a two-player game, $\mathcal{G} = (S, E, w)$ its underlying graph, $s_{\mathsf{init}} \in S$ the initial state, $\lambda_2^f \in \Lambda_2^F$ a finite-memory stochastic model of $\mathcal{P}_2$, and $(\mu = \frac{a}{b}, \nu) \in \mathbb{Q}^2$ a pair of thresholds, with $a \in \mathbb{Z}$ and $b \in \mathbb{N}_0$. Then $\mathcal{P}_1$ has a satisfying strategy for the BWC mean-payoff problem in G if and only if $\mathcal{P}_1$ has a satisfying strategy when considering the thresholds pair $(0, \nu' = b \cdot \nu - a)$ and the weight function w' such that $\forall e \in E$, $w'(e) = b \cdot w(e) - a$.*

Proof. Recall the mean-payoff of a play is defined as the (infimum) limit of the mean weight over prefixes of increasing length. Hence, the affine transformation applied on weights carries over to play values: for all $\pi \in \mathsf{Plays}(\mathcal{G})$, we have that $\mathsf{MP}_{w'}(\pi) = b \cdot \mathsf{MP}_w(\pi) - a$, where the subscript denotes which weight function we consider. Let $\lambda_1 \in \Lambda_1$ be a strategy of $\mathcal{P}_1$. First, consider the worst-case requirement of eq. (1). We claim that

$$\forall \lambda_2 \in \Lambda_2, \forall \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2), \mathsf{MP}_w(\pi) > \mu = \frac{a}{b} \iff \forall \lambda_2 \in \Lambda_2, \forall \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2), \mathsf{MP}_{w'}(\pi) > 0.$$

This is trivial by applying the affine transformation. Second, consider the expected value requirement of eq. (2). We claim that

$$\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1, \lambda_2^f]}(\mathsf{MP}_w) > \nu \iff \mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1, \lambda_2^f]}(\mathsf{MP}_{w'}) > \nu' = b \cdot \nu - a.$$

The expected value operator is well-known to be linear. Thus, by extracting the affine transformation, we obtain that $\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1, \lambda_2^f]}(\mathsf{MP}_{w'}) = b \cdot \mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1, \lambda_2^f]}(\mathsf{MP}_w) - a$, which proves our point and concludes the proof. □

Worst-case winning. Second, we prove in Lemma 3 that a necessary condition for a strategy to satisfy the BWC problem is to avoid visiting states that are losing for the worst-case mean-payoff requirement. This justifies lines 3-7 of the algorithm. Observe that the graph of $G^w = G^i \restriction S_{WC}$ contains no deadlock as otherwise it would contradict the fact that $\mathcal{P}_1$ can satisfy the worst-case threshold problem from states in S_{WC} in the game G^i. Also note that $G^w[\lambda_2^i]$ remains a well-defined MDP as there exists no edge from states $s \in S_2^i \cap S_{WC}$ to states $s' \in S \setminus S_{WC}$, otherwise $\mathcal{P}_2$ could win the game from s by reaching s', by prefix-independence of the mean-payoff, and so s would no belong to S_{WC}.

Lemma 3. *Let $G = (\mathcal{G}, S_1, S_2)$ be a two-player game, $\mathcal{G} = (S, E, w)$ its underlying graph, $s_{\mathsf{init}} \in S$ the initial state, $\lambda_2^f \in \Lambda_2^F$ a finite-memory stochastic model of $\mathcal{P}_2$, and $\nu \in \mathbb{Q}$ the expected value threshold. Let $S_{WC} \subseteq S$ be the set of winning states for $\mathcal{P}_1$ for the worst-case threshold problem. If $\lambda_1 \in \Lambda_1^F$ is a satisfying strategy for the BWC mean-payoff problem, then*

$$\forall \lambda_2 \in \Lambda_2, \forall \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2), \forall s \in S \setminus S_{WC}, s \notin \pi. \tag{3}$$

Proof. Formally, let $S_{WC} := \{s \in S \mid \exists \lambda_1 \in \Lambda_1, \forall \lambda_2 \in \Lambda_2, \forall \pi \in \mathsf{Outs}_G(s, \lambda_1, \lambda_2), \mathsf{MP}(\pi) > 0\}$. We claim that a winning strategy of $\mathcal{P}_1$ for the BWC problem must avoid states that are not in S_{WC} and prove it by contradiction. Indeed, let $\lambda_1 \in \Lambda_1^F$ be such a strategy. Assume eq. (3) is false: there exist $\lambda_2 \in \Lambda_2$, $\pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2)$ and $s \in S \setminus S_{WC}$ such that $s \in \pi$. By definition of S_{WC} and determinacy of mean-payoff games [13,25], it is the case that in state s, $\mathcal{P}_2$ has a winning strategy for the worst-case objective: there exists $\lambda_2' \in \Lambda_2$ such that for all $\lambda_1' \in \Lambda_1$, there exists $\pi' \in \mathsf{Outs}_G(s, \lambda_1', \lambda_2')$ such that $\neg(\mathsf{MP}(\pi') > 0)$, i.e., $\mathsf{MP}(\pi') \leq 0$. Hence, consider the strategy λ_2'' of $\mathcal{P}_2$ that plays according to λ_2 up to the first visit of s and according to λ_2' afterwards. Clearly, there exists an outcome $\pi'' \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2'')$ such that $\pi'' = \rho \cdot \pi'$ and $\mathsf{MP}(\pi') \leq 0$ for some $\rho \in \mathsf{Prefs}(\mathcal{G})$. Since the mean-payoff objective is prefix-independent, we have that $\mathsf{MP}(\pi'') \leq 0$. Thus, the strategy λ_1 of $\mathcal{P}_1$ does not satisfy eq. (1), which contradicts the hypothesis and concludes the proof. □

Memoryless stochastic model. Finally, we show in Lemma 4 that we can study the equivalent BWC problem on the game obtained by product of the original game and the Moore machine of the stochastic model, using a memoryless stochastic model instead of the finite-memory one (lines 8-10 of BWC_MP). Having a memoryless stochastic model proves useful in the main steps of the algorithm (lines 11-17) as it guarantees that the game G and the MDP $G[\lambda_2^{\mathsf{stoch}}]$ possess the same underlying graph.

Lemma 4. *Let $G = (\mathcal{G}, S_1, S_2)$ be a two-player game, $\mathcal{G} = (S, E, w)$ its underlying graph, $s_{\mathsf{init}} \in S$ the initial state, $\lambda_2^f \in \Lambda_2^F(G)$ a finite-memory stochastic model of $\mathcal{P}_2$, $\mathcal{M}(\lambda_2^f) = (\mathsf{Mem}, \mathsf{m}_0, \alpha_\mathsf{u}, \alpha_\mathsf{n})$ its Moore machine, and $\nu \in \mathbb{Q}$ the expected value threshold. Let $G' = G \otimes \mathcal{M}(\lambda_2^f)$ be the product game and $\lambda_2^m \in \Lambda_2^M(G')$ the memoryless transcription of λ_2^f on G'. The two following statements are equivalent.*

(a) $\mathcal{P}_1$ has a strategy to satisfy the BWC problem on G against the finite-memory stochastic model λ_2^f.
(b) $\mathcal{P}_1$ has a strategy to satisfy the BWC problem on G' against the memoryless stochastic model λ_2^m.

Proof. We define the product game $G' = G \otimes \mathcal{M}(\lambda_2^f) = (\mathcal{G}', S_1', S_2')$, with $\mathcal{G}' = (S', E', w')$, as follows.

- $S' = S \times \mathsf{Mem}$, $S_1' = S_1 \times \mathsf{Mem}$, $S_2' = S_2 \times \mathsf{Mem}$;
- $\forall s_1, s_2 \in S, \forall \mathsf{m}_1, \mathsf{m}_2 \in \mathsf{Mem}, ((s_1, \mathsf{m}_1), (s_2, \mathsf{m}_2)) \in E' \Leftrightarrow (s_1, s_2) \in E \wedge \alpha_\mathsf{u}(\mathsf{m}_1, s_1) = \mathsf{m}_2$;
- $\forall e = ((s_1, \mathsf{m}_1), (s_2, \mathsf{m}_2)) \in E'$, $w'(e) = w((s_1, s_2))$;
- $s'_{\mathsf{init}} = (s_{\mathsf{init}}, \mathsf{m}_0)$ is the new initial state.

Given the finite-memory stochastic model $\lambda_2^f \in \Lambda_2^F(G)$ of $\mathcal{P}_2$, we transcript it into a memoryless strategy $\lambda_2^m \in \Lambda_2^M(G')$ on the product game such that

$$\forall s_1 \in S_2, \forall ((s_1, \mathsf{m}_1), (s_2, \mathsf{m}_2)) \in E', \lambda_2^m((s_1, \mathsf{m}_1))((s_2, \mathsf{m}_2)) = \alpha_\mathsf{n}(\mathsf{m}_1, s_1)(s_2).$$

Basically, we have integrated the finite memory of λ_2^f into the states of G' and defined the remaining corresponding memoryless strategy λ_2^m on G'.

We first prove that *(a)* $\Rightarrow$ *(b)*. Assume $\lambda_1 \in \Lambda_1^F(G)$ is a satisfying strategy for the BWC problem on G, i.e., it satisfies eq. (1) and (2) against the stochastic model λ_2^f. We build a corresponding strategy $\lambda_1' \in \Lambda_1^F(G')$ that is winning against λ_2^m in G' as follows.

$$\forall \rho' = (s_0, \mathsf{m}_0)(s_1, \mathsf{m}_1) \dots (s_k, \mathsf{m}_k) \in \mathsf{Prefs}_1(G'), (s_{k+1}, \mathsf{m}_{k+1}) \in S' \text{ such that } ((s_k, \mathsf{m}_k), (s_{k+1}, \mathsf{m}_{k+1})) \in E',$$
$$\lambda_1'(\rho')((s_{k+1}, \mathsf{m}_{k+1})) = \lambda_1(\mathsf{proj}_S(\rho'))(s_{k+1}).$$

Note that strategy λ_1' is well-defined as λ_1 is. Consider the worst-case requirement of the BWC problem on G', eq. (1). Let $\pi' \in \mathsf{Outs}_{G'}(s'_{\mathsf{init}}, \lambda_1')$ be any outcome consistent with the newly defined strategy λ_1'. By definition of λ_1', we have that $\pi = \mathsf{proj}_S(\pi') \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1)$ is an outcome consistent with λ_1 in G. Hence, by hypothesis, we have that $\mathsf{MP}(\pi) > 0$. By definition of w', we have that $\mathsf{MP}_{w'}(\pi') = \mathsf{MP}_w(\pi)$, thus $\mathsf{MP}(\pi') > 0$ and the worst-case requirement is satisfied. Now consider the expected value, eq. (2). By hypothesis, we have that $\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1, \lambda_2^f]}(\mathsf{MP}) > \nu$. We claim that $\mathbb{E}_{s'_{\mathsf{init}}}^{G'[\lambda_1', \lambda_2^m]}(\mathsf{MP}) > \nu$. Indeed, remark that there is a bijection between outcomes in $\mathsf{Outs}_{G[\lambda_1, \lambda_2^f]}(s_{\mathsf{init}})$

and $\mathsf{Outs}_{G'[\lambda_1',\lambda_2^m]}(s'_{\mathsf{init}})$ using the projection operator because the memory update function α_u of the Moore machine $\mathcal{M}(\lambda_2^f)$ is deterministic. As the values of plays are preserved by the changes to w' and the probability measures of cylinder sets are also preserved by definition of λ_2^m, the claim is verified.

Second, we show that *(b)* $\Rightarrow$ *(a)*. Assume $\lambda_1' \in \Lambda_1^F(G')$ is a satisfying strategy for the BWC problem on G', against the stochastic model λ_2^m. We build a corresponding strategy $\lambda_1 \in \Lambda_1^F(G)$ that is winning against λ_2^f in G. Thanks to the update function of $\mathcal{M}(\lambda_2^f)$ being deterministic, given a prefix $\rho = s_0 s_1 \dots s_k \in \mathsf{Prefs}(G)$, there is a unique corresponding prefix $\rho' \in \mathsf{Prefs}(G')$ such that $\rho = \mathsf{proj}_S(\rho')$. Hence we define λ_1 as follows.

$$\forall \rho = s_0 s_1 \dots s_k \in \mathsf{Prefs}_1(G), s_{k+1} \in S \text{ such that } (s_k, s_{k+1}) \in E,$$
$$\lambda_1(\rho)(s_{k+1}) = \lambda_1'(\rho')((s_{k+1}, \mathsf{m}_{k+1})),$$

with $\rho' = (s_0, \mathsf{m}_0)(s_1, \mathsf{m}_1)\dots(s_k, \mathsf{m}_k)$ the unique prefix in $\mathsf{Prefs}_1(G')$ such that $\rho = \mathsf{proj}_S(\rho')$ and $\mathsf{m}_{k+1} = \alpha_u(\mathsf{m}_k, s_k)$. Strategy λ_1 is well-defined. Consider the worst-case requirement. Let $\pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1)$ be any consistent outcome and π' the unique corresponding play in G'. By definition, $\pi' \in \mathsf{Outs}_{G'}(s'_{\mathsf{init}}, \lambda_1')$. Hence, by construction, $\mathsf{MP}(\pi) = \mathsf{MP}(\pi')$, and by hypothesis, $\mathsf{MP}(\pi') > 0$, which proves that λ_1 satisfies eq. (1) in G. Finally, consider the expected value requirement, eq. (2). Again, there is a bijection between $\mathsf{Outs}_{G[\lambda_1,\lambda_2^f]}(s_{\mathsf{init}})$ and $\mathsf{Outs}_{G'[\lambda_1',\lambda_2^m]}(s'_{\mathsf{init}})$. Since weights and probability measures are preserved, we have that $\mathbb{E}^{G[\lambda_1,\lambda_2^f]}_{s_{\mathsf{init}}}(\mathsf{MP}) = \mathbb{E}^{G'[\lambda_1',\lambda_2^m]}_{s'_{\mathsf{init}}}(\mathsf{MP}) > \nu$. This concludes our proof that assertions *(a)* and *(b)* are equivalent. □

Remark that given a satisfying strategy in the product game, the proof of Lemma 4 describes how to obtain a corresponding satisfying strategy in the original game. Hence, strategies obtained through algorithm BWC_MP are preserved alongside the answer to the BWC problem when going back to the original game.

We now study the main steps of algorithm BWC_MP, assuming the simplifying assumptions granted by the preprocessing.

4.4 Classification of end-components

In the following, we consider a game G where all states are winning for the worst-case requirement (eq. (1)), a memoryless stochastic model $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^M$, and $P = G[\lambda_2^{\mathsf{stoch}}]$, the MDP obtained when this stochastic model is followed by $\mathcal{P}_2$. As sketched in Sect. 4.1, the crux of the algorithm is the analysis of the end-components of P.

Long-run appearance of end-components. Lemma 5 recalls a well-known property of MDPs: under any arbitrary strategy of $\mathcal{P}_1$, the set of states visited infinitely often along a play almost-surely constitutes an EC. Notice the abuse of notation as discussed in Sect. 2.

Lemma 5 ([9,10]). *Let $P = (\mathcal{G}, S_1, S_\Delta, \Delta)$ be an MDP, $\mathcal{G} = (S, E, w)$ its underlying graph, $\mathcal{E} \subseteq 2^S$ the set of its end-components, $s_{\mathsf{init}} \in S$ the initial state, and $\lambda_1 \in \Lambda_1(P)$ an arbitrary strategy of $\mathcal{P}_1$. Then, we have that*

$$\mathbb{P}^{P[\lambda_1]}_{s_{\mathsf{init}}}\left(\{\pi \in \mathsf{Outs}_{P[\lambda_1]}(s_{\mathsf{init}}) \mid \mathsf{Inf}(\pi) \in \mathcal{E}\}\right) = 1.$$

Hence the expected value $\mathbb{E}^{P[\lambda_1]}_{s_{\mathsf{init}}}(\mathsf{MP})$ depends exclusively on the values of end-components (because the mean-payoff of any play belongs to $[-W, W]$ and thus the value of plays not entering ECs, which set has probability measure zero, cannot be infinite).

Winning end-components. First, we introduce some notations. We respectively denote G_Δ and P_Δ the game and MDP where the underlying graph is limited to the subset of edges which are assigned non-zero probability by $\Delta = \lambda_2^{\mathsf{stoch}}$, i.e., $E_\Delta \subseteq E$. By definition, ECs are computed with regard to E_Δ, hence the ECs of P_Δ are exactly equal to the ECs of P. Moreover, we have that

$$\Lambda_2(G_\Delta) = \left\{\lambda_2 \in \Lambda_2(G) \mid \forall \rho \cdot s \in \mathsf{Prefs}_2(G), \mathsf{Supp}(\lambda_2(\rho \cdot s)) \subseteq \mathsf{Supp}(\Delta(s))\right\},$$

whereas available choices are unchanged for $\mathcal{P}_1$ in G_Δ as edges in $E \setminus E_\Delta$ all start in states of $\mathcal{P}_2$.

Using these notations, we now define *winning* ECs as the ECs $U \in \mathcal{E}$ where $\mathcal{P}_1$ can ensure satisfaction of the worst-case requirement in the corresponding subgame $G_\Delta \restriction U$. Note that we consider G_Δ as we only need to consider strategies of $\mathcal{P}_2$ that share the support of the stochastic model. That is because our goal is to ensure that $\mathcal{P}_1$ can benefit from (the maximal expectation achievable in) these ECs, which is only safe with regard to the worst-case requirement if $\mathcal{P}_1$ can force that all outcomes that may occur when facing the stochastic model yield a strictly positive mean-payoff. Note that reacting to an arbitrary strategy of $\mathcal{P}_2$, i.e., a strategy in $\Lambda_2(G)$, as required in eq. (1), will be considered in the following sections: for now we only care about $\lambda_2^{\mathsf{stoch}}$ and satisfaction of the expected value requirement as specified in eq. (2).

Definition 2. *Let $G = (\mathcal{G}, S_1, S_2)$ be a two-player game, $\mathcal{G} = (S, E, w)$ its underlying graph, $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^M$ a memoryless stochastic model of $\mathcal{P}_2$, $P = G[\lambda_2^{\mathsf{stoch}}] = (\mathcal{G}, S_1, S_\Delta = S_2, \Delta = \lambda_2^{\mathsf{stoch}})$ the resulting MDP and G_Δ the game reduced to non-zero probability edges. Let $U \in \mathcal{E}$ be an end-component of P. Then, we have that*

– *$U \in \mathcal{W}$, the* winning ECs, *if*

$$\exists \lambda_1 \in \Lambda_1(G_\Delta \restriction U), \forall \lambda_2 \in \Lambda_2(G_\Delta \restriction U), \forall s \in U, \forall \pi \in \mathsf{Outs}_{(G_\Delta \restriction U)}(s, \lambda_1, \lambda_2), \mathsf{MP}(\pi) > 0\,; \tag{4}$$

– *$U \in \mathcal{L}$, the* losing ECs, *otherwise. By determinacy of mean-payoff games,*

$$\exists \lambda_2 \in \Lambda_2(G_\Delta \restriction U), \forall \lambda_1 \in \Lambda_1(G_\Delta \restriction U), \exists s \in U, \exists \pi \in \mathsf{Outs}_{(G_\Delta \restriction U)}(s, \lambda_1, \lambda_2), \mathsf{MP}(\pi) \leq 0. \tag{5}$$

Note that an EC is winning if $\mathcal{P}_1$ has a worst-case winning strategy from *all* states. This point is important as it may well be the case that winning strategies exist in a strict subset of states of the EC. This does not contradict the definition of ECs as strongly connected subgraphs, as the latter only guarantees that every state can be reached *with probability one*, and not necessarily surely. Hence one cannot call upon the prefix-independence of the mean-payoff to extend the existence of a winning strategy to all states. Such a situation can be observed on the game of Fig. 3, where the EC U_2 is losing (because from s_1, the outcome $(s_1 s_3 s_4)^\omega$ can be forced by $\mathcal{P}_2$, yielding mean-payoff $-1/3 \leq 0$), while its sub-EC U_3 is winning. From s_1, $\mathcal{P}_1$ can ensure to reach U_3 almost-surely, but not surely, which is critical in this case.

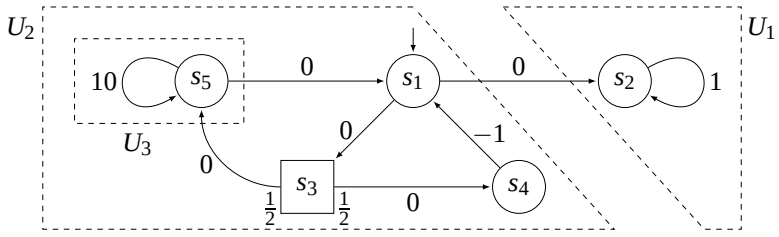

Fig. 3: End component U_2 is losing. The set of maximal winning ECs is $\mathcal{U}_{\mathrm{W}} = \mathcal{W} = \{U_1, U_3\}$.

Maximality. As discussed in Sect. 4.1, we can restrict our analysis to *maximal* winning ECs in the following. This is a consequence of Lemma 6.

Lemma 6. *Let $U_1, U_2 \in \mathcal{W}$ be two winning ECs in the MDP P such that $U_1 \subsetneq U_2$. Let v_1^*, v_2^* denote the respective maximal expected values achievable by $\mathcal{P}_1$ in U_1 and U_2. Then, we have that $v_1^* \leq v_2^*$.*

Proof. First notice that the maximal expectation achievable in an EC does not depend on the starting state inside the EC. Hence, assume any state $s_2 \in U_2$. Since $U_1 \subsetneq U_2$ and U_2 is an end-component, $\mathcal{P}_1$ can reach a state in U_1 with probability one from s_2. Since the mean-payoff value function only takes finite values, the contribution of plays that do not reach U_1 in the expected value is null. Finally, by prefix-independence of the mean-payoff, we can forget about the finite prefixes outside U_1 and we deduce that $v_2^* \geq v_1^*$. □

We formally define the set of *maximal winning ECs* as

$$\mathcal{U}_{\mathrm{w}} = \{U \in \mathcal{W} \mid \forall U' \in \mathcal{W}, U \subseteq U' \Rightarrow U = U'\}.$$

Remark that this set is not to be confused with the set of winning maximal ECs, $\{U \in \mathcal{W} \mid \forall U' \in \mathcal{E}, U \subseteq U' \Rightarrow U = U'\} \subseteq \mathcal{U}_{\mathrm{w}}$, which holds no particular interest for us. While the total number of winning ECs $|\mathcal{W}| \leq |\mathcal{E}| \leq 2^{|S|}$ can be exponential in the number of states of the game, the number of maximal winning ECs $|\mathcal{U}_{\mathrm{w}}| \leq |S|$ is bounded by this number of states, as all ECs of $\mathcal{U}_{\mathrm{w}}$ are disjoint (because the union of two winning ECs is itself a winning EC). Hence, restriction to the maximal winning ECs is a cornerstone in the overall NP$\cap$coNP complexity that we claim for algorithm BWC_MP.

Illustration. Consider the running example in Fig. 2. Note that states s_1, s_2 and s_5 do not belong to any EC: given any strategy of $\mathcal{P}_1$ in P, with probability one, any consistent outcome will only visit those states a finite number of times (Lemma 5). The set of *maximal winning ECs* is $\mathcal{U}_{\mathrm{w}} = \{U_2, U_3\}$. Obviously, those ECs are disjoint. The set of winning ECs is much larger, $\mathcal{W} = \mathcal{U}_{\mathrm{w}} \cup \{\{s_9, s_{10}\}, \{s_{10}, s_{11}\}, \{s_6, s_7\}, \{s_6, s_8\}\}$.

End-component U_1 is *losing*. Indeed, in the subgame $G_\Delta \restriction U_1$, the strategy consisting in always picking the -1 edge guarantees an outcome which mean-payoff is negative. Note that this edge is present in E_Δ as it is assigned probability $1/2$ by the stochastic model. Here, we witness why it is important to base our definition of winning ECs on the game G_Δ rather than G. Indeed, in $G \restriction U_2$, $\mathcal{P}_2$ can also guarantee a negative mean-payoff by always choosing edges with weight -1. However, to achieve this, $\mathcal{P}_2$ has to pick edges that are in E_Δ: this will never happen against the stochastic model and as such, this can be watched by $\mathcal{P}_1$ to see if $\mathcal{P}_2$ uses an arbitrary antagonistic strategy, and dealt with. If $\mathcal{P}_2$ conforms to E_Δ, i.e., if he plays in G_Δ, he has to pick the edge of weight 1 in s_7 and $\mathcal{P}_1$ has a worst-case winning strategy consisting in always choosing to go in s_7. This EC is thus classified as *winning*. Note that for U_3, in both subgames $G \restriction U_3$ and $G_\Delta \restriction U_3$, $\mathcal{P}_1$ can guarantee a strictly positive mean-payoff by playing $(s_9\, s_{10})^\omega$: even *arbitrary* strategies of $\mathcal{P}_2$ cannot endanger $\mathcal{P}_1$ in this case.

Lastly, consider the game depicted in Fig. 3. While U_2 is a strict superset of U_3, the former is losing whereas the latter is winning, as explained above. Hence, the set $\mathcal{U}_{\mathrm{w}}$ is equal to $\{U_1, U_3\}$.

Computation of the maximal winning end-components. Obviously, from a complexity standpoint, to benefit from the polynomial size of $\mathcal{U}_{\mathrm{w}}$, in contrast to the potentially exponential size of $\mathcal{W}$, we need to compute $\mathcal{U}_{\mathrm{w}}$ without first computing all winning ECs of $\mathcal{W}$. We present an algorithm to do so, called MWEC, in Alg. 2. Lemma 7 establishes that MWEC is correct and complete, and is in NP$\cap$coNP, resorting to an NP$\cap$coNP oracle to solve mean-payoff games (i.e., decide the answer of the worst-case threshold problem), other than that implementing polynomial operations. It operates under the assumption that all edges are of non-zero probability, i.e., $E_\Delta = E$. This is w.l.o.g. as it suffices to remove those edges as a preprocessing step (they have no impact in the definitions of ECs and winning ECs).

Lemma 7. *Let $P = (\mathcal{G}, S_1, S_\Delta, \Delta)$ be an MDP, with $\mathcal{G} = (S, E, w)$ its underlying graph such that $E_\Delta = E$. Then algorithm* MWEC *computes its set of maximal winning ECs $\mathcal{U}_{\mathrm{w}} = \mathrm{MWEC}(P)$ and is in NP$\cap$coNP.*

Algorithm MWEC can be sketched as follows. Given a non-empty MDP, it first computes its decomposition into maximal end-components[8] (without distinction between winning and losing ECs). This can be obtained in polynomial time [7]. Afterwards, it checks for each of these ECs if it is winning or not, in the sense of Def. 2. If the EC is winning, it is now part of the set of claimed maximal winning ECs, denoted $\mathcal{M}_{\mathrm{w}}$ in the algorithm, and the algorithm will not recurse on this set of states. If the EC is losing, then it may still be the case that a sub-EC is winning, as discussed in Sect. 4.4. Hence, the algorithm eliminates all worst-case losing states and executes recursively on the induced subgame. It stops recursing on sub-MDPs whenever one is declared winning or empty. Since it suppresses at least one state in each call, the algorithm is ensured to stop. Moreover, the number of calls is polynomial in the size of the MDP. Deciding if an EC is winning or losing requires calling an NP$\cap$coNP oracle solving the worst-case threshold problem [30,23,17].

[8] Given an MDP P with a set of ECs $\mathcal{E}$, an EC U is said to be *maximal* in P if for all $U' \in \mathcal{E}$, $U \subseteq U' \Rightarrow U = U'$. This is not to be confused with the definition of maximal winning ECs, given in this section. In particular, maximal ECs need not be winning in general, whereas maximal winning ECs need not be maximal ECs in the sense we just defined (they only need to be maximal with regard to other *winning* ECs).

Algorithm 2 MWEC(P)

Require: $P = (\mathcal{G}, S_1, S_\Delta, \Delta)$, with $\mathcal{G} = (S, E, w)$ such that $E_\Delta = E$
Ensure: $\mathcal{M}_\mathrm{W} = \mathcal{U}_\mathrm{W}$, the set of maximal winning ECs of P

1: **if** P is empty **then**
2: **return** $\emptyset$
3: **else**
4: Compute $\mathcal{M}_\mathrm{W} := \{U_1, \ldots, U_n\}$ the maximal EC decomposition of P
5: **for all** $i = 1, \ldots, n$ **do**
6: Compute $L_i \subseteq U_i$ the set of states from which $\mathcal{P}_2$ has a strategy to enforce $\mathsf{MP} \leq 0$ in the subgame $G \restriction U_i$, i.e.,

$$L_i := \left\{ s \in U_i \mid \forall \lambda_1 \in \Lambda_1(P \restriction U_i), \exists \pi \in \mathsf{Outs}_{P \restriction U_i}(s, \lambda_1), \mathsf{MP}(\pi) \leq 0 \right\}$$

7: **if** $L_i \neq \emptyset$ **then**
8: $\mathcal{M}_\mathrm{W} := (\mathcal{M}_\mathrm{W} \setminus \{U_i\}) \cup \mathrm{MWEC}(P \restriction (U_i \setminus L_i))$
9: **return** $\mathcal{M}_\mathrm{W}$

The remainder of this section is dedicated to the proof of the correctness and completeness of the algorithm, as well as an illustration of its operation. We first state several remarks on its functioning that will be of importance in the proof.

Remark 2. In line 6, we have that $P \restriction U_i$ is a well-defined MDP, since U_i is an EC. Similarly, in line 8, $P \restriction (U_i \setminus L_i)$ is also a well-defined MDP. Indeed, from all states $s \in (U_i \setminus L_i) \cap S_1$, there exists an edge from s that goes to a state of $U_i \setminus L_i$, otherwise s would be a losing state (and so would be in L_i). Moreover, for all states $s \in (U_i \setminus L_i) \cap S_\Delta$, there is no edge from s that goes in L_i (otherwise s would be in L_i) nor in $S \setminus U_i$ (otherwise U would not be an EC), and therefore the probability distribution $\Delta(s)$ is still well-defined on $P \restriction (U_i \setminus L_i)$. Additionally, this sub-MDP still verifies that there is no edge with probability zero.

Remark 3. Let U be an EC of P, L its set of losing states (as computed by line 6), and $V \subseteq U \setminus L$. Then V is a winning EC of $P \restriction (U \setminus L)$ if and only if V is a winning EC of P. This follows from the same reasoning as for Rem. 2.

Remark 4. Let U be a winning EC of P, strictly included in a losing EC V of P. Let $s \in V$ be a worst-case losing state (i.e., a state from which $\mathcal{P}_1$ cannot guarantee a strictly positive mean-payoff in the subgame $G \restriction V$, as defined at line 6 of the algorithm). Then we claim that $s \notin U$. Indeed, suppose the contrary. From s, $\mathcal{P}_2$ can enforce $\mathsf{MP} \leq 0$ against any strategy $\lambda_1 \in \Lambda_1(P \restriction V)$, and a fortiori could do so against any strategy $\lambda_1 \in \Lambda_1(P \restriction U)$ (notice that only $\mathcal{P}_1$ can decide to leave U as U is an EC). Therefore, U would not be winning.

Proof. We first show that the algorithm is *sound*, i.e., $\mathcal{M}_\mathrm{W} \subseteq \mathcal{U}_\mathrm{W}$. It is done by induction on the size of P. If P is empty, the claim is clear. Otherwise let $U \in \mathcal{M}_\mathrm{W}$. There are two cases.

1. U is equal to some U_i computed at line 4 and has never been removed from $\mathcal{M}_\mathrm{W}$. It means that L_i is empty, and by definition of L_i, that U_i is winning. Also, U_i is trivially a *maximal* winning EC as it belongs to the maximal EC decomposition of P.
2. U has been added at line 8 as the result of some recursive call $\mathrm{MWEC}(P \restriction (U_i \setminus L_i))$ for some maximal EC U_i of P. Since $L_i \neq \emptyset$, the set $U_i \setminus L_i$ is strictly smaller than S, and by induction hypothesis, U is a winning EC of $P \restriction (U_i \setminus L_i)$. By Rem. 3, it is also a winning EC of P. It remains to show that U is a *maximal* winning EC of P. Suppose that it is not the case. Then there exists a strict superset U' of U which is a winning EC of P. Clearly, $U' \subseteq U_i$ since U_i is a maximal EC of P, and maximal ECs are pairwise disjoint. Morevoer, U' is a subset of $U_i \setminus L_i$ by Rem. 4. By Rem. 3, it is therefore a winning EC of $P \restriction (U_i \setminus L_i)$, which contradicts the maximality of U in $P \restriction (U_i \setminus L_i)$.

We now establish that the algorithm is *complete*, i.e., $\mathcal{U}_\mathrm{W} \subseteq \mathcal{M}_\mathrm{W}$. Again, it is proved by induction on the size of P. If P is empty, then the claim is obviously true. Now, suppose that P is non-empty, and let $U \in \mathcal{U}_\mathrm{W}$. There are two cases.

1. U is a maximal EC of P. In that case, it will be computed at line 4 and never removed from $\mathcal{M}_\mathrm{W}$ (because the set of losing states will be empty as U is winning).

2. U is not a maximal EC in P. Therefore there exists some maximal EC U_i of P, which is losing and strictly contains U. Let L_i be the non-empty set of worst-case losing states of U_i (as computed by line 6). We have to show that U is a maximal winning EC of $P \restriction (U_i \setminus L_i)$, in which case we could conclude by induction hypothesis, i.e., U would be returned by the recursive call $\mathrm{MWEC}(P \restriction (U_i \setminus L_i))$. By Rem. 4, U and L_i are disjoint, and therefore $U \subseteq (U_i \setminus L_i)$. By Rem. 2 and Rem. 3, U is a winning EC of $P \restriction (U_i \setminus L_i)$, since it is a winning EC of P. It remains to show that U is a *maximal* winning EC of $P \restriction (U_i \setminus L_i)$. Suppose that there exists a strict superset U' of U such that U' is a winning EC of $P \restriction (U_i \setminus L_i)$. By Rem. 3, U' would also be a winning EC of P, which contradicts that $U \in \mathcal{U}_\mathrm{w}$ by definition of $\mathcal{U}_\mathrm{w}$ as the set of maximal winning ECs. It implies that U is a maximal winning EC of $P \restriction (U_i \setminus L_i)$ and thus that $U \in \mathcal{M}_\mathrm{w}$.

Finally, consider the complexity class of this algorithm. Assume that we have an NP $\cap$ coNP oracle solving the worst-case threshold problem [30,23,17] and called in line 6. The number of recursive calls to MWEC is linear in $|S|$, the number of states of P, because in the loop of line 5, the sets U_i are pairwise disjoint, and because each call is executed after eliminating at least one state. Moreover, the maximal EC decomposition of P can be computed in $O(|S|^2)$ [7]. Overall, we thus obtain that MWEC belongs to NP $\cap$ coNP. □

Consider the execution of algorithm MWEC on the MDP described in Fig. 3. In its first call, it computes the maximal EC decomposition $\mathcal{M}_\mathrm{w} = \{U_1, U_2\}$. Now, for U_1, we have that L_1 is empty and thus U_1 remains in $\mathcal{M}_\mathrm{w}$. On the contrary, for U_2, we have that $L_2 = \{s_1, s_3, s_4\}$. Hence the algorithm suppresses U_2 from $\mathcal{M}_\mathrm{w}$ and recurses on the sub-MDP $P \restriction (U_2 \setminus L_2) = P \restriction \{s_5\}$. There, the maximal EC decomposition gives the unique EC U_3 which is winning since $L_3 = \emptyset$, and thus remains in $\mathcal{M}_\mathrm{w}$. The algorithm ends with $\mathcal{M}_\mathrm{w} = \{U_1, U_3\}$. Clearly we have that $\mathcal{M}_\mathrm{w} = \mathcal{U}_\mathrm{w}$ as proved before.

4.5 Winning end-components are almost-surely reached in the long-run

Recall that Lemma 5 states that under any arbitrary strategy $\lambda_1 \in \Lambda_1$, the set of infinitely visited states of the outcome of the MDP $P[\lambda_1] = G[\lambda_1, \lambda_2^{\mathsf{stoch}}]$ is almost-surely equal to an EC. In this section, we refine this result and show that under any *finite-memory* strategy $\lambda_1^f \in \Lambda_1^F$ *satisfying the BWC problem*, the set of infinitely visited states is almost-surely equal to a *winning* EC. In other words, the long-run probability of *negligible states*, defined as $S_{\mathsf{neg}} = S \setminus \bigcup_{U \in \mathcal{U}_\mathrm{w}} U = S \setminus \bigcup_{U \in \mathcal{W}} U$, is zero.

Lemma 8. *Let $G = (\mathcal{G}, S_1, S_2)$ be a two-player game, $\mathcal{G} = (S, E, w)$ its underlying graph, $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^M$ a memoryless stochastic model of $\mathcal{P}_2$, $P = G[\lambda_2^{\mathsf{stoch}}]$ the resulting MDP and $s_{\mathsf{init}} \in S$ the initial state. Let $\lambda_1^f \in \Lambda_1^F$ be a finite-memory strategy of $\mathcal{P}_1$ that satisfies the BWC problem for thresholds $(0, \nu) \in \mathbb{Q}^2$. Then, we have that*

$$\mathbb{P}_{s_{\mathsf{init}}}^{P[\lambda_1^f]}\left(\left\{\pi \in \mathsf{Outs}_{P[\lambda_1^f]}(s_{\mathsf{init}}) \mid \mathsf{Inf}(\pi) \in \mathcal{W}\right\}\right) = 1. \tag{6}$$

Proof. Let $\lambda_1^f \in \Lambda_1^F$ be a finite-memory BWC satisfying strategy. By Lemma 5, we have that eq. (6) is verified for $U \in \mathcal{E} = \mathcal{W} \cup \mathcal{L}$. It remains to show that the probability of having a losing EC, i.e., an EC in $\mathcal{L}$, is zero.

By contradiction, assume there exists some $U_\mathrm{L} \in \mathcal{L}$ such that

$$\mathbb{P}_{s_{\mathsf{init}}}^{P[\lambda_1^f]}\left(\left\{\pi \in \mathsf{Outs}_{P[\lambda_1^f]}(s_{\mathsf{init}}) \mid \mathsf{Inf}(\pi) = U_\mathrm{L}\right\}\right) > 0. \tag{7}$$

Since λ_1^f is finite-memory, we have that $M = P[\lambda_1^f]$ is a *finite* MC. Thus, we consider the bottom strongly-connected components (BSCCs) of M and eq. (7) implies that some outcomes of $\mathsf{Outs}_M(s_{\mathsf{init}})$ will be trapped in a BSCC corresponding to U_L (i.e., this BSCC is reachable with non-zero probability in M), and visit all its states infinitely often. Since U_L is losing, this BSCC induces plays where the mean-payoff is not strictly positive. Indeed, strategy $\lambda_2^{\mathsf{stoch}}$ of $\mathcal{P}_2$ suffices to produce consistent outcomes that are worst-case losing thanks to Def. 2 (as only the support matters for the worst-case requirement, not the exact probabilities). By prefix-independence of the mean-payoff value function, we obtain the existence of plays of M, starting in s_{init}, and inducing a mean-payoff that does not satisfy the worst-case threshold. This shows that λ_1^f is not winning for the BWC problem and by contradiction, concludes our proof. □

The direct consequence of this statement is that *edges involving negligible states do not contribute to the overall expectation* of finite-memory strategies satisfying the BWC problem. Based on this observation, we propose in Sect. 4.8 a modification of the MDP $P = G[\lambda_2^{\mathsf{stoch}}]$ that will help us synthesize satisfying strategies when they exist.

Remark 5. The proof of Lemma 8 relies on the finite memory of the strategy. Similar reasoning cannot be applied if the strategy of $\mathcal{P}_1$ uses infinite memory. Indeed, the Markov chain $M = P[\lambda_1^f]$ becomes infinite and we cannot base our analysis on BSCCs anymore (as they need not exist in general, outcomes cannot be trapped in a BSCC). As a matter of fact, we prove in Sect. 4.11 that infinite-memory strategies may benefit from negligible states forming losing ECs in some cases, by staying in them forever with a non-zero probability, and thus it is not possible to neglect them in the overall expectation.

Illustration. Consider U_1 on Fig. 2. By Def. 2, this EC is losing as always taking the edge of weight -1 is a winning strategy for $\mathcal{P}_2$ in $G_\Delta \downharpoonright U_1$. The optimal expectation achievable in $P \downharpoonright U_1$ by $\mathcal{P}_1$ is 4: this is higher than what is achievable in both U_2 and U_3. Note that there exists no winning EC included in U_1. By Lemma 5, we know that any strategy of $\mathcal{P}_1$ will see its expectation bounded by the maximum between the optimal expectations of the ECs U_1, U_2 and U_3. Lemma 8 further refines this bound by restricting it to the maximum between the expectations of U_2 and U_3. Indeed, it states that $\mathcal{P}_1$ cannot benefit from the expected value of U_1 while using finite memory, as being trapped in U_1 with non-zero probability induces the existence of outcomes losing for the worst-case (here, outcomes that always take the -1 edge). Since U_1 neither helps for the worst-case nor for the expectation, there is no point in playing inside it and $\mathcal{P}_1$ may as well cross it directly and try to maximize its expectation using the winning ECs, U_2 and U_3. The set of negligible states in P is $S_{\mathsf{neg}} = S \setminus (U_2 \cup U_3) = \{s_1, s_2, s_3, s_4, s_5\}$.

In the game depicted in Fig. 3, we already observed that $\mathcal{E} = \{U_1, U_2, U_3\}$, $\mathcal{W} = \mathcal{U}_{\mathrm{w}} = \{U_1, U_3\}$ and $\mathcal{L} = \{U_2\}$. Consider the negligible state $s_1 \in S_{\mathsf{neg}} = U_2 \setminus U_3$. As a consequence of Lemma 8, we have that a finite-memory strategy of $\mathcal{P}_1$ may only take the edge (s_1, s_3) finitely often in order to ensure the worst-case requirement. Indeed, the losing outcome $(s_1 s_3 s_4)^\omega$ would exist (while of probability zero) if $\mathcal{P}_1$ were to play this edge infinitely often. Therefore, it is clear that $\mathcal{P}_1$ can only ensure that U_3 is reached with a probability arbitrarily close to one, and not equal to one, because at some point, he has to switch to edge (s_1, s_2) (after a bounded time since $\mathcal{P}_1$ uses a finite-memory strategy).

4.6 Winning end-component with non-zero probabilities: combined strategy

In this section, we take a closer look at what happens inside a winning EC *where the stochastic model assigns non-zero probabilities* to all possible edges. We will show how to deal with edges of probability zero in Sect. 4.7. For the sake of readability, we make Assumption 1. Obviously, similar reasoning can be applied to all the such winning ECs in a larger game.

Assumption 1. *Let $G = (\mathcal{G}, S_1, S_2)$ be a two-player game, $\mathcal{G} = (S, E, w)$ its underlying graph, $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^M$ a memoryless stochastic model of $\mathcal{P}_2$, and $P = G[\lambda_2^{\mathsf{stoch}}] = (\mathcal{G}, S_1, S_\Delta = S_2, \Delta = \lambda_2^{\mathsf{stoch}})$ the resulting MDP. We assume that G is reduced to a unique maximal winning EC, i.e., $\mathcal{U}_{\mathrm{w}} = \{S\}$, and that $G_\Delta = G$, i.e., the set of edges of probability zero, $E \setminus E_\Delta$, is empty.*

Our claim is that inside such a winning EC, $\mathcal{P}_1$ has a *finite-memory* strategy that simultaneously (a) ensures the worst-case requirement, and (b) yields an expected value which is ε-close to the maximal expectation of the EC. Consequently, we establish Theorem 2 and Corollary 1.

Theorem 2. *Let $G = (\mathcal{G}, S_1, S_2)$ be a two-player game reduced to a unique winning EC, $\mathcal{G} = (S, E, w)$ its underlying graph, $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^M$ a memoryless stochastic model of $\mathcal{P}_2$ such that $E_\Delta = E$, $P = G[\lambda_2^{\mathsf{stoch}}] = (\mathcal{G}, S_1, S_\Delta = S_2, \Delta = \lambda_2^{\mathsf{stoch}})$ the resulting MDP, $s_{\mathsf{init}} \in S$ an initial state and $v^* \in \mathbb{Q}$ the maximal expected value achievable by $\mathcal{P}_1$ in P. Then, for all $\varepsilon > 0$, there exists a finite-memory strategy of $\mathcal{P}_1$ that satisfies the BWC problem for the thresholds pair $(0, v^* - \varepsilon)$.*

The remainder of this section is dedicated to the proof of Thm. 2. It is a surprisingly positive result as it essentially states that $\mathcal{P}_1$ can *guarantee both* the worst-case and the expected value thresholds *without sacrifying any performance* (in terms of play values) except for some arbitrarily small ε. The key idea is to build a finite-memory strategy based

on careful alternation between two memoryless strategies: one which is optimal for the worst-case, and one which is optimal for the expected value. The proof requires deep understanding of the limiting properties of Markov chains, such as the rate of convergence towards a stationary distribution. Nevertheless, we provide an intuitive sketch of the combined strategy and illustrate it on the running example in the following.

Corollary 1. *In a game G reduced to a winning EC, $\mathcal{P}_1$ has a strategy for the BWC problem for thresholds $(0,\nu)$ against a stochastic model $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^M$ such that $E_\Delta = E$ if and only if the optimal expected value in $G[\lambda_2^{\mathsf{stoch}}]$ is strictly greater than ν.*

Proof. Consider the left-to-right implication. Assume λ_1 is the BWC strategy. By definition, the optimal expected value ν^* is at least equal to $\mathbb{E}_{s_{\mathsf{init}}}^{P[\lambda_1]}(\mathsf{MP})$, which is strictly greater than ν by hypothesis. Now consider the converse implication. Let ν^* be the optimal expected value. By hypothesis, $\nu^* > \nu$. By Thm. 2, for any $\varepsilon > 0$, there exists a strategy $\lambda_1 \in \Lambda_1^F$ that satisfies the BWC problem for thresholds $(0, \nu^* - \varepsilon)$. In particular, it is possible to choose $\varepsilon \leq \nu^* - \nu$ to obtain the claim and conclude the proof. □

Individual requirements - memoryless strategies. First, consider the two requirements - worst-case and expected value - independently. By definition of winning ECs (eq. (4)) and the hypothesis that $G = G_\Delta$, $\mathcal{P}_1$ has a strategy to guarantee a strictly positive mean-payoff against any strategy of the adversary in the game. As discussed in Sect. 2, pure memoryless optimal strategies exist for $\mathcal{P}_1$ [24,13]. In the following, we denote $\lambda_1^{wc} \in \Lambda_1^{PM}$ such a strategy, and μ^* the optimal mean-payoff value, i.e., the minimal mean-payoff that λ_1^{wc} ensures against any strategy of $\mathcal{P}_2$. Hence, we have that

$$\mu^* = \inf_{\lambda_2' \in \Lambda_2} \left\{ \mathsf{MP}(\pi) \mid \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1^{wc}, \lambda_2') \right\} = \sup_{\lambda_1 \in \Lambda_1} \inf_{\lambda_2' \in \Lambda_2} \left\{ \mathsf{MP}(\pi) \mid \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2') \right\} > 0,$$

since we are in a winning EC for the worst-case threshold $\mu = 0$.

Similarly, we define a pure memoryless strategy maximizing the expected value in the MDP $P = G[\lambda_2^{\mathsf{stoch}}]$ induced by applying the memoryless stochastic model $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^M$ on G. We denote this strategy $\lambda_1^e \in \Lambda_1^{PM}$, and we write the associated expectation as $\nu^* = \mathbb{E}_{s_{\mathsf{init}}}^{P[\lambda_1^e]}(\mathsf{MP})$. Notice that we manipulate equivalently strategies on the game and on the MDP thanks to their shared underlying graph (Sect. 4.3). The existence of a pure memoryless optimal strategy follows from [14] (see Sect. 2). However, we here require, *without loss of generality*, that λ_1^e is chosen[9] in order to satisfy an additional property: we want that the Markov chain $P[\lambda_1^e]$ be *unichain*, i.e., containing a unique recurrent class (i.e., a unique bottom strongly-connected component when considering edges which are assigned non-zero probability in the MC), and possibly some transient states. This will be useful later to apply needed technical results (Lemma 9). As evoked, it is always possible to choose such a strategy. Intuitively, $\mathcal{P}_2$ cannot force the existence of multiple recurrent classes in the MC as it would contradict the fact that we are inside an EC (by definition, $\mathcal{P}_1$ must be able to force the visit of any state with probability one). Hence, it remains to argue that $\mathcal{P}_1$ has no interest in inducing multiple recurrent classes, as he cannot increase the expected value by doing so. This is clear since either the different recurrent classes yield the same expectation, in which case one suffices, or they yield different expectations, in which case an optimal strategy like λ_1^e will only use the one that produces the maximal expectation ($\mathcal{P}_1$ has the power to restrict the MC to the class of his choice by definition of EC).

In general, one cannot hope to satisfy the BWC problem by following only strategy λ_1^{wc} or only strategy λ_1^e, although they suffice when their respective requirements are considered independently. Fortunately, it is possible to build upon those two strategies in order to achieve simultaneous satisfaction with a combined finite-memory strategy.

Defining a combined strategy. Based on the existence of strategies λ_1^{wc} and λ_1^e, we define a pure finite-memory strategy $\lambda_1^{cmb} \in \Lambda_1^{PF}$ that carefully and dynamically alternates between the two memoryless strategies to ensure satisfaction of the BWC problem. Our strategy is *parameterized by two naturals*: K and L.

Definition 3. *In a game G satisfying Assumption 1, we define the* combined strategy $\lambda_1^{cmb} \in \Lambda_1^{PF}$ *as follows.*

(a) Play λ_1^e for K steps[10] *and memorize $\mathsf{Sum} \in \mathbb{Z}$, the sum of weights encountered during these K steps.*

[9] There may exist several pure memoryless optimal strategies, all yielding the same expected value, by definition of optimality.
[10] By step we mean taking any edge in the game, be it from a state of $\mathcal{P}_1$ or $\mathcal{P}_2$.

(b) If $\mathsf{Sum} > 0$*, then go to (a).*
Else, play λ_1^{wc} *during* L *steps then go to (a).*

We define *periods* as sequences played from the beginning of a phase of type *(a)* or *(b)* up to its end, i.e., the beginning of a new period. Intuitively, in a period of type *(a)*, the strategy mimics the optimal expectation strategy. By playing λ_1^e long enough, we can ensure that the mean-payoff obtained during the period is very close to ν^*, with probability close to one (Lemma 9). Still, we need to ensure the worst-case threshold in all cases. This may in general not be ensured by periods of type *(a)*. Hence, we keep a memory of the running sum of weights in the period. This requires a finite number of bits of memory as the sum takes values in $\{-K \cdot W, -K \cdot W + 1, \ldots, K \cdot W - 1, K \cdot W\}$. If $\mathsf{Sum} > 0$ after the K steps, then the mean-payoff over the period satisfies the worst-case requirement (eq. (1)) and the strategy can immediately start a new period of type *(a)*. Otherwise, it is necessary to compensate in order to satisfy the worst-case requirement. In that case, the strategy mimics the optimal worst-case strategy λ_1^{wc} for L steps. Such a strategy guarantees that cycles in the outcome have a strictly positive sum of weights since $\mu^* > 0$, as discussed before. As Sum is lower bounded after K steps, there exists a value of L such that the total sum of weights (and thus the mean-payoff) over periods *(a)* + *(b)* is strictly positive. Using the fact that all weights are integers, we further deduce that the sum over a period is at least equal to one. By the boundedness of the length of a period, we thus prove that the overall mean-payoff along a play stays *strictly* positive. Hence λ_1^{cmb} satisfies eq. (1).

While this sketch is sufficient to see that the worst-case requirement is satisfied by strategy λ_1^{cmb}, proving that we can choose K and L such that its expected value is ε-close to ν^* is more involved. Intuitively, periods of type *(b)* must not happen too frequently, nor be too long, in order to have a boundable impact on the overall expectation. The cornerstone to achieve such a moderate impact of periods of type *(b)* resides in the fact that a linear increase in K produces an exponential decrease in the need for a period of type *(b)* (Lemma 9) whereas it only requires a linear increase in L to ensure the worst-case requirement (see Def. 4 and Lemma 10). Note that the need for a decreasing contribution of periods of type *(b)* to the overall expectation explains why we need to track the current sum Sum and cannot settle for a simpler strategy that would play periods *(a)* and *(b)* in strict alternation (cf. Rem. 7).

In a nutshell, we claim that under Assumption 1, it is always possible to find values for constants K and L such that strategy λ_1^{cmb} satisfies the BWC problem for $(0, \nu^* - \varepsilon)$, as stated in Theorem 2. Before proving it, we illustrate the combined strategy and introduce some intermediary technical results. Notice that implementing λ_1^{cmb} only requires finite memory as strategies λ_1^e and λ_1^{wc} are memoryless, constants K and L have finite values, and Sum takes a finite number of values.

Illustration. Consider the subgame $G_\Delta \restriction U_3 = G \restriction U_3$ of the game in Fig. 2 and the initial state $s_{\mathsf{init}} = s_{10}$. Clearly, the worst-case requirement can be satisfied, that is why the EC is classified as winning. Indeed, always choosing to go to s_9 when in s_{10} is an optimal memoryless worst-case strategy λ_1^{wc} that guarantees a mean-payoff $\mu^* = 1$. The expectation of this strategy is $\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^{wc}, \lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) = 1$. On the other hand, the strategy λ_1^e that always selects the edge going to s_{11} is optimal regarding the expected value criterion: it induces an expectation[11] $\nu^* = \Big(0 + \big(1/2 \cdot 9 + 1/2 \cdot (-1)\big)\Big)/2 = 2$ against the stochastic model $\lambda_2^{\mathsf{stoch}}$. However, it can only guarantee a mean-payoff of value $-1/2$ in the worst-case.

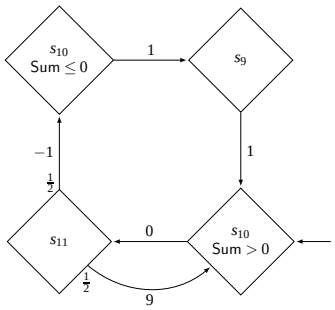

Fig. 4: Markov chain $G[\lambda_1^{cmb}, \lambda_2^{\mathsf{stoch}}]$ induced by the combined strategy λ_1^{cmb} and the stochastic model $\lambda_2^{\mathsf{stoch}}$ over the winning EC U_3 of G.

By Theorem 2, we know that it is possible to find finite-memory strategies satisfying the BWC problem for any thresholds pair $(0, 2-\varepsilon)$, $\varepsilon > 0$. In particular, consider the thresholds pair $(0, 3/2)$. We build a combined strategy λ_1^{cmb} as described in Def. 3. Let $K = L = 2$: the strategy plays the edge (s_{10}, s_{11}) once, then if the edge of value 9 has been chosen by $\mathcal{P}_2$, it chooses (s_{10}, s_{11}) again; otherwise it chooses the edge (s_{10}, s_9) once and then resumes choosing (s_{10}, s_{11}). This strategy satisfies the BWC problem. In the worst-case, $\mathcal{P}_2$ always chooses the -1 edge, but each time he does so, the -1 is followed by two $+1$ thanks to the cycle $s_{10}s_9s_{10}$.

[11] Given an irreducible MC $M = (\mathcal{G}, \delta)$, with $\mathcal{G} = (S, E, w)$, one can compute its limiting stationary distribution by finding the unique probability vector $\mathbf{v} \in [0,1]^{|S|}$ such that $\mathbf{v}\mathbf{P}_\delta = \mathbf{v}$, where $\mathbf{P}_\delta$ denotes the transition matrix derived from δ. The expected mean-payoff value can then be obtained by multiplying the row vector $\mathbf{v}$ by the column vector $\mathbf{e} \in \mathbb{R}^{|S|}$ that contains the respective expected weights over outgoing edges for each state. That is: $\forall s \in S$, $\mathbf{e}(s) = \sum_{s' \in S} \delta(s)(s') \cdot w((s, s'))$, and $\mathbb{E}_s^M = \mathbf{v} \cdot \mathbf{e}$.

Strategy λ_1^{cmb} hence guarantees a mean-payoff equal to $(0-1+1+1)/4 = 1/4 > 0$ in the worst-case. For the expected value requirement, we can build the Markov chain $G[\lambda_1^{cmb}, \lambda_2^{\mathsf{stoch}}]$ (Fig. 4) and check that its expectation is $\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^{cmb}, \lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) = 5/3 > 3/2$.

Remark 6. Memoryless strategies do not suffice for the BWC problem, even with randomization. Indeed, the edge (s_{10}, s_{11}) cannot be assigned a non-zero probability as it would endanger the worst-case requirement (since the outcome $(s_{10}s_{11})^\omega$ cycling on the edge of weight -1 would exist and have a negative mean-payoff). Hence, the only acceptable memoryless strategy is λ_1^{wc}, which has only an expectation of 1.

Technical results. Before proving the correctness of strategy λ_1^{cmb}, we need to introduce an important property verified by the Markov chain $G[\lambda_1^e, \lambda_2^{\mathsf{stoch}}]$. It is well-known that in the long-run, the probability of outcomes that induce a mean-payoff equal to the expectation of the MC is one. Lemma 9 shows that, for sufficiently long prefixes, it is possible to bound the probability of having a mean-payoff which differs from the expected value by more than a given $\varepsilon > 0$. In particular, it implies that for sufficiently large values of K, this probability decreases exponentially with K. This will help us bound the impact of periods of type *(b)* on the overall expectation.

Lemma 9 (Follows from the extension of [19, Thm. 2] proposed in [27]). *For all initial state s_{init}, for all $\varepsilon > 0$, for some constants $c_1, c_2 > 0$, there exists $K_0 \in \mathbb{N}$ such that, for all $K \geq K_0$,*

$$\mathbb{P}_{s_{\mathsf{init}}}^{G[\lambda_1^e, \lambda_2^{\mathsf{stoch}}]}\Big(\pi \in \mathsf{Plays}(G[\lambda_1^e, \lambda_2^{\mathsf{stoch}}]) \;\Big|\; \big|\mathsf{MP}(\pi(K)) - \nu^*\big| \geq \varepsilon\Big) \leq \mathcal{F}(K, \varepsilon) = \frac{c_1}{e^{c_2 \cdot K \cdot \varepsilon^2}}.$$

In [19, Thm. 2], Glynn and Ormoneit present an extension of Hoeffding's inequality [21] for uniformly ergodic Markov chains. Straight application of this result in our setting is not possible, as the MC $G[\lambda_1^e, \lambda_2^{\mathsf{stoch}}]$ does not need to be aperiodic in general. Nevertheless, this result is extended to unichain MCs (possibly periodic) in [27]. Hence, Lemma 9 reformulates the latter result in our precise context. Notice that the MC $G[\lambda_1^e, \lambda_2^{\mathsf{stoch}}]$ is unichain thanks to the choice of λ_1^e, as presented earlier.

Proof. For the sake of completeness, we sketch the main steps proposed by Tracol [27] to extend the results of [19]. Consider the MC $G[\lambda_1^e, \lambda_2^{\mathsf{stoch}}]$: it can be decomposed in a set of transient states and a unique recurrent class, i.e., a bottom strongly-connected component. Assume this BSCC is periodic, of period $d \in \mathbb{N}_0$. We can decompose it in d aperiodic classes on which we are able to apply the bound provided by [19, Thm. 2]. The key idea is then to obtain a unified bound by aggregating the bounds obtained for each aperiodic MC, given sufficiently large values of the constants.

A word on the constants of Lemma 9. Careful analysis of the proofs of [19, Thm. 2] and [27, Prop. 2] reveals that c_1 is exponential in ε and polynomial in the characteristics of the MC, while c_2 is only polynomial in the characteristics of the MC. More importantly, constant K_0 is polynomial in the size of the MC and polynomial in the largest weight W (exponential in its encoding). □

Analysis of the combined strategy. Our goal is to show that for any $\varepsilon > 0$, there exist two naturals K and L such that λ_1^{cmb} proves the correctness of Theorem 2. First, we define L as a *linear* function of K. Note that the main purpose of K is to create periods of type *(a)* long enough to have an expected mean-payoff close to the optimal value achieved by λ_1^e, i.e., ν^*. The aim of L is for periods of type *(b)* to be long enough to compensate the possible negative effect of periods of type *(a)* and thus ensure the worst-case requirement. As stated before, L should not grow too quickly to preserve an overall mean-payoff which is mainly influenced by periods of type *(a)* (hence close to the optimal expectation ν^*).

Definition 4. *Given a natural constant $K \in \mathbb{N}$, we define* $L = \left\lfloor \dfrac{K \cdot W + |S| \cdot W + |S| \cdot \mu^*}{\mu^*} \right\rfloor + 1.$

Remark 7. Obviously, L needs to be proportional to K to preserve the worst-case requirement as the amount to compensate is bounded by $K \cdot W$. Consequently, we can observe the need for the bookkeeping of Sum in strategy λ_1^{cmb} in contrast to a non-dynamic alternation scheme between periods of type *(a)* and *(b)*. Indeed, in a strategy following the latter scheme, the long-term expectation would be close to $\frac{K \cdot \nu^* + L \cdot \mu^*}{K+L}$. As L is not constant but proportional to K, one can easily see that this expression does not tend to ν^* when K tends to infinity (which is required when ε tends to zero according to Lemma 9). Thus, strict alternation does not suffice to satisfy the thresholds pair presented in Thm. 2.

Under the value of L given in Def. 4, Lemma 10 states that the worst-case requirement is satisfied. The idea is to decompose any play into an infinite sequence of periods, each of them having a bounded length and ensuring a strictly positive sum of weights, thus yielding an overall strictly positive mean-payoff of the play.

Lemma 10. *For any $K \in \mathbb{N}$, the combined strategy $\lambda_1^{cmb} \in \Lambda_1^{PF}$ is such that*

$$\forall \lambda_2 \in \Lambda_2, \forall \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1^{cmb}, \lambda_2), \mathsf{MP}(\pi) > 0.$$

Proof. Let λ_2 be an arbitrary strategy of $\mathcal{P}_2$ and π any outcome in $\mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1^{cmb}, \lambda_2)$. By definition of λ_1^{cmb}, we decompose the play in a sequence of periods of type *(a)* and *(b)*. That is, $\pi = \rho_0\rho_1\rho_2\dots$ where, for all $i \geq 0$, ρ_i is a finite sequence of states that is either of length K if ρ_i is of type *(a)* or of length L if ρ_i is of type *(b)*. Moreover, ρ_0 is of type *(a)* and for all $i \geq 1$, ρ_i is of type *(b)* if and only if ρ_{i-1} is of type *(a)* and such that $\mathsf{Sum}(\rho_{i-1}) \leq 0$ (i.e., the sum of weights along the sequence is not strictly positive). We regroup each sequence of type *(b)* with its predecessor of type *(a)* and obtain $\pi = \rho'_0\rho'_1\rho'_2\dots$ such that all sequences are either of type *(a)* or of type *(a)*$+$*(b)*.

Consider any sequence ρ'_i of type *(a)*. Since it is not followed by a period of type *(b)*, we know that $\mathsf{Sum}(\rho'_i) > 0$. Now consider any sequence ρ'_i of type *(a)*$+$*(b)*. At the end of the period of type *(a)*, the sum is bounded from below by $-K \cdot W$. During the period of type *(b)*, an optimal *memoryless* worst-case strategy λ_1^{wc} is followed and consequently, all formed cycles have a mean-payoff at least equal to μ^*. Hence, the sum of weights over the period of type *(b)* is at least $(L - |S|) \cdot \mu^* - |S| \cdot W$. By Def. 4, this induces that the overall sum over the sequence of type *(a)*$+$*(b)* is $\mathsf{Sum}(\rho'_i) > 0$.

In both cases, we thus have that $\mathsf{Sum}(\rho'_i) > 0$. But since weights are integers, this implies a stronger inequality: $\mathsf{Sum}(\rho'_i) \geq 1$. We go back to the play seen as an infinite sequence of states $\pi = s_0 s_1 s_2 \dots$. Thanks to the previous observations, we can now state that the total sum of weights up to any state s_t, $t \geq 0$, is bounded by

$$\mathsf{TP}(\pi(t)) \geq -\big[t \bmod (K+L)\big] \cdot W + \left\lfloor \frac{t}{K+L} \right\rfloor \cdot 1 \geq -(K+L) \cdot W + \frac{t}{K+L} - 1.$$

Hence the mean-payoff of the play π is

$$\mathsf{MP}(\pi) \geq \liminf_{t \to \infty} \left[\frac{-(K+L) \cdot W}{t} + \frac{1}{K+L} - \frac{1}{t} \right] = \frac{1}{K+L} > 0,$$

which concludes the proof. □

It remains to show that for any $\varepsilon > 0$, there exists a value $K \in \mathbb{N}$ such that the expected value requirement (eq. (2)) is also satisfied by λ_1^{cmb}. This is proved in Lemma 11. Again, decomposition of plays into periods needs to be considered. Furthermore, the crux of the proof resides in the use of Lemma 9: it induces that when the length of periods of type *(a)* grows linearly, the probability of periods of type *(b)* decreases exponentially. Since the length of periods of type *(b)* only grows linearly in K, the overall impact of periods *(b)* in the expectation tends to zero when K tends to infinity. Hence, the expectation of λ_1^{cmb} tends to the optimal expectation v^* and classical convergence analysis provides the Lemma.

Lemma 11. *For any $\varepsilon > 0$, there exists $K \in \mathbb{N}$ such that $\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^{cmb}, \lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) > v^* - \varepsilon$.*

Proof. For the proof, we assume that $\varepsilon \leq v^*$, otherwise the claim is obviously true for any $K \in \mathbb{N}$ since the mean-payoff of any outcome consistent with λ_1^{cmb} is strictly positive by application of Lemma 10. Now, for a given $K \in \mathbb{N}$, consider the corresponding finite MC $M(K) = G[\lambda_1^{cmb}, \lambda_2^{\mathsf{stoch}}]$ where λ_1^{cmb} is defined with the parameter K (i.e. periods of type *(a)* are of length K). To obtain the claim, we prove that $\mathbb{E}_{s_{\mathsf{init}}}^{M(K)}(\mathsf{MP}) \xrightarrow[K \to \infty]{} v^*$. Similarly to the proof of Lemma 10, any outcome of $M(K)$ is an outcome consistent with λ_1^{cmb} in G and thus can be decomposed in an infinite sequence of periods of types *(a)* and *(a)*$+$*(b)*.

To begin with, we will consider (i) the expectation over *one* period of type *(a)*, and (ii) the expectation over *one* period of type *(a)*$+$*(b)*, as well as the respective probabilities of seeing such periods whenever a new period begins. Note that we can do that because periods of type *(a)* and *(a)*$+$*(b)* are independent events by definition of λ_1^{cmb}. Hence,

the probability that some period has a specific type, and the expected value of that period, are not influenced by what happened over previous periods.[12]

(i) Let us first consider periods of type *(a)*. By Def. 3, $\mathcal{P}_1$ follows the strategy λ_1^e during those periods. Recall that $\mathbb{E}_{s_{\text{init}}}^{G[\lambda_1^e,\lambda_2^{\text{stoch}}]}(\mathsf{MP}) = v^*$. Let us denote by $e_{(a)}$ the expected mean-payoff over a period of type *(a)*. By Lemma 9, for any $\varepsilon > 0$, there exists $K_0 \in \mathbb{N}$ such that for all $K \geq K_0$, this expected value is bounded from below by

$$e_{(a)} \geq (1 - \mathcal{F}(K,\varepsilon)) \cdot (v^* - \varepsilon) + \mathcal{F}(K,\varepsilon) \cdot x,$$

with x a lower bound on *any* consistent outcome. Since any period of type *(a)* is not followed by a period of type *(b)* (otherwise it would be considered as a period of type *(a)+(b)*), we know that the sum of weights along the K steps of the period is greater than or equal to 1, and therefore we can take $x \geq 1/K$ (cf. proof of Lemma 10).

(ii) Now, consider periods of type *(a)+(b)*. As shown in the proof of Lemma 10, we know that the expected mean-payoff over such a period, denoted by $e_{(a)+(b)}$, is greater than or equal to $1/(K+L)$. Now consider the probability $p_{(a)+(b)}$ that a period is of type *(a)+(b)*. By definition of strategy λ_1^{cmb}, $p_{(a)+(b)}$ is equal to the probability of having a total sum of weights less than or equal to 0 after K steps of playing according to λ_1^e. Since we assume that $\varepsilon \leq v^*$ and since $v^* \geq \mu^* > 0$, we can bound from above this probability by the probability to have a mean-payoff less than $v^* - \varepsilon$ over the K steps. Repeating the argument of point (i) and applying Lemma 9, we deduce that $p_{(a)+(b)} \leq \mathcal{F}(K,\varepsilon)$ for all $K \geq K_0$.

We now have sufficient arguments to study the overall expected value over m periods. Let $p_{(a)}$ denote the probability of periods of type *(a)*. Note that $p_{(a)} = 1 - p_{(a)+(b)}$. The expected number of periods of type *(a)* is $m \cdot p_{(a)}$. Similarly, $m \cdot p_{(a)+(b)}$ is the expected number of periods of type *(a)+(b)*. The expected sum of weights over the m periods is therefore $m \cdot p_{(a)} \cdot e_{(a)} \cdot K + m \cdot p_{(a)+(b)} \cdot e_{(a)+(b)} \cdot (K+L)$ since periods of type *(a)* have length K and periods of type *(a)+(b)* length $K+L$. The expected length of the m periods is $m \cdot p_{(a)} \cdot K + m \cdot p_{(a)+(b)} \cdot (K+L)$. Finally, we have that

$$\mathbb{E}_{s_{\text{init}},\, m\,\text{periods}}^{M(K)}(\mathsf{MP}) = \frac{m \cdot p_{(a)} \cdot e_{(a)} \cdot K + m \cdot p_{(a)+(b)} \cdot e_{(a)+(b)} \cdot (K+L)}{m \cdot p_{(a)} \cdot K + m \cdot p_{(a)+(b)} \cdot (K+L)}. \tag{8}$$

Clearly, this expression does not depend on the number of periods m. This is consistent with our analysis since we have established that periods are statistically independent. Also, note that this reasoning is only correct for complete periods. Nonetheless, any prefix $\pi(n)$, $n \geq 0$, of the play is composed of a sequence of complete periods followed by a suffix of length bounded by $(K+L)$ and of total sum of weights bounded by $-(K+L) \cdot W$ and $(K+L) \cdot W$. Hence, we characterize the expected mean-payoff over the n first steps of a play as

$$\frac{\left\lfloor \frac{n}{l} \right\rfloor \cdot l \cdot \mathbb{E}_{s_{\text{init}},\,1\,\text{period}}^{M(K)}(\mathsf{MP}) - (K+L) \cdot W}{n} \leq \mathbb{E}_{s_{\text{init}},\,n\,\text{steps}}^{M(K)}(\mathsf{MP}) \leq \frac{\left\lfloor \frac{n}{l} \right\rfloor \cdot l \cdot \mathbb{E}_{s_{\text{init}},\,1\,\text{period}}^{M(K)}(\mathsf{MP}) + (K+L) \cdot W}{n},$$

with $l = p_{(a)} \cdot K + p_{(a)+(b)} \cdot (K+L)$, the expected length of a period. Naturally, the finite suffix proves to be negligible when n grows, hence

$$\mathbb{E}_{s_{\text{init}}}^{M(K)}(\mathsf{MP}) = \liminf_{n \to \infty} \left[\mathbb{E}_{s_{\text{init}},\,n\,\text{steps}}^{M(K)}(\mathsf{MP}) \right] = \mathbb{E}_{s_{\text{init}},\,1\,\text{period}}^{M(K)}(\mathsf{MP}). \tag{9}$$

Remark that eq. (9) uses the equality between the expectation over the values of plays and the limit of the expectation over values of prefixes. This equality is verified for the mean-payoff value function but does not need to be true for arbitrary value functions.

Back to eq. (8), with $m = 1$, we use $e_{(a)+(b)} \geq 1/(K+L) > 0$ and assume $K > 0$ to obtain

$$\mathbb{E}_{s_{\text{init}}}^{M(K)}(\mathsf{MP}) \geq \frac{p_{(a)} \cdot e_{(a)}}{p_{(a)} + p_{(a)+(b)} \cdot \left(\frac{K+L}{K}\right)}.$$

Again, we assume K large enough to ensure $p_{(a)} > 0$ (such a K exists by consequence of Lemma 9) and get

$$\mathbb{E}_{s_{\text{init}}}^{M(K)}(\mathsf{MP}) \geq \frac{e_{(a)}}{1 + \left(\frac{p_{(a)+(b)}}{p_{(a)}}\right) \cdot \left(\frac{K+L}{K}\right)}.$$

[12] For the sake of simplicity, we omit that different periods do not necessarily start in the same state, as the resulting impact on the expectation is negligible for sufficiently long periods.

By (i) and (ii), we have $p_{(a)} \geq 1 - \mathcal{F}(K,\varepsilon)$, $p_{(a)+(b)} \leq \mathcal{F}(K,\varepsilon)$, and $e_{(a)} \geq (1-\mathcal{F}(K,\varepsilon)) \cdot (\nu^* - \varepsilon)$. We derive that

$$\mathbb{E}^{M(K)}_{s_{\mathsf{init}}}(\mathsf{MP}) \geq \frac{(1-\mathcal{F}(K,\varepsilon)) \cdot (\nu^* - \varepsilon)}{1 + \frac{\mathcal{F}(K,\varepsilon)\cdot(K+L)}{(1-\mathcal{F}(K,\varepsilon))\cdot K}}. \tag{10}$$

Recall that ultimately, we want to prove that $\mathbb{E}^{M(K)}_{s_{\mathsf{init}}}(\mathsf{MP}) \xrightarrow[K\to\infty]{} \nu^*$. Consider what happens when $K \to \infty$ in eq. (10): notice that L is linear in K by Def. 4, hence $L \to \infty$, and that $\mathcal{F}(K,\varepsilon) \to 0$ by Lemma 9. This does not suffice to conclude on the possible convergence of the lower bound given in eq. (10). The crux of the argument is given by Lemma 9: $\mathcal{F}(K,\varepsilon)$ decreases exponentially for a linear increase in K. Thus, we have that $\mathcal{F}(K,\varepsilon) \cdot (K+L) \to 0$. Therefore,

$$\lim_{K\to\infty} \left[\frac{(1-\mathcal{F}(K,\varepsilon)) \cdot (\nu^* - \varepsilon)}{1 + \frac{\mathcal{F}(K,\varepsilon)\cdot(K+L)}{(1-\mathcal{F}(K,\varepsilon))\cdot K}} \right] = \nu^* - \varepsilon. \tag{11}$$

Notice two facts. First, for any $K \in \mathbb{N}$, we have that $\mathbb{E}^{M(K)}_{s_{\mathsf{init}}}(\mathsf{MP}) \leq \mathbb{E}^{G[\lambda_1^e, \lambda_2^{\mathsf{stoch}}]}_{s_{\mathsf{init}}}(\mathsf{MP}) = \nu^*$ by the optimality of λ_1^e. Second, eq. (11) is valid for any ε such that $\nu^* \geq \varepsilon > 0$. Hence we observe that the sequence of expected values $(\mathbb{E}^{M(K)}_{s_{\mathsf{init}}}(\mathsf{MP}))_{K\geq 0}$ is bounded from above and from below by two sequences converging to ν^*. Ergo

$$\mathbb{E}^{M(K)}_{s_{\mathsf{init}}}(\mathsf{MP}) \xrightarrow[K\to\infty]{} \nu^*.$$

By definition of convergence, for any ε such that $\nu^* \geq \varepsilon > 0$, there exists $K \in \mathbb{N}$ such that $\mathbb{E}^{M(K)}_{s_{\mathsf{init}}}(\mathsf{MP}) > \nu^* - \varepsilon$, which concludes our proof. □

Based on Lemma 10 and Lemma 11, Theorem 2 follows straightforwardly and concludes our analysis of winning ECs with no edges of probability zero. We will present how we can extend these results to arbitrary winning ECs in the next section.

4.7 Starting in a winning end-component: witness-and-secure strategy

We now turn to winning ECs with potential existence of edges of probability zero, i.e., $E_\Delta \subseteq E$. We present in this section how to construct a finite-memory strategy that can benefit ε-closely from the maximal expectation achievable in such winning ECs when facing the stochastic model $\lambda_2^{\mathsf{stoch}}$, while guaranteeing satisfaction of the worst-case requirement even against *arbitrary* strategies of $\mathcal{P}_2$ (i.e., strategies that may use edges in $E \setminus E_\Delta$). It is crucial to notice that we now consider *a complete game*, i.e., not necessarily reduced to a single winning EC as in Sect. 4.6. Still, we assume that the play starts in a winning EC: consistent outcomes will stay in it when $\mathcal{P}_2$ follows $\lambda_2^{\mathsf{stoch}}$ (because $\mathcal{P}_1$ will have no interest to leave), but may exit the EC if $\mathcal{P}_2$ takes edges of probability zero, either by the action of $\mathcal{P}_2$ (recall there may exist edges that leave the EC in $E \setminus E_\Delta$) or the action of $\mathcal{P}_1$ (which may need to leave to guarantee a strictly positive mean-payoff).

Theorem 3. *Let $G = (\mathcal{G}, S_1, S_2)$ be a two-player game, $\mathcal{G} = (S, E, w)$ its underlying graph, $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^M$ a memoryless stochastic model of $\mathcal{P}_2$, $P = G[\lambda_2^{\mathsf{stoch}}] = (\mathcal{G}, S_1, S_\Delta = S_2, \Delta = \lambda_2^{\mathsf{stoch}})$ the resulting MDP, $U \in \mathcal{W}$ a winning EC, $s_{\mathsf{init}} \in U$ an initial state inside the EC, and $\nu^* \in \mathbb{Q}$ the maximal expected value achievable by $\mathcal{P}_1$ in $P \restriction U$. Then, for all $\varepsilon > 0$, there exists a finite-memory strategy of $\mathcal{P}_1$ that satisfies the BWC problem for the thresholds pair $(0, \nu^* - \varepsilon)$.*

We prove this theorem in the following. Let us first give some key intuition. With respect to the expected value requirement of eq. (2), the hypothesis is that $\mathcal{P}_2$ *will* follow its stochastic model $\lambda_2^{\mathsf{stoch}}$. Hence, he will only play edges in E_Δ and the combined strategy proposed in Sect. 4.6 suffices to achieve the claimed expectation and guarantee the worst-case threshold against $\lambda_2^{\mathsf{stoch}}$ (basically, we can apply Thm. 2 on G_Δ). Now, we still have to account for arbitrary strategies of $\mathcal{P}_2$ to satisfy the worst-case requirement of eq. (1). Notice that the combined strategy suffices to ensure it against all strategies playing exclusively in E_Δ. So, it only remains to deal with strategies that chooses some edges in $E \setminus E_\Delta$. It is easy for $\mathcal{P}_1$ to *witness* if $\mathcal{P}_2$ chooses an edge outside E_Δ (as the stochastic model is assumed known

by $\mathcal{P}_1$). If this happens, $\mathcal{P}_1$ can *secure* its mean-payoff value by switching from the combined strategy to a worst-case winning strategy, which exists in all states due to the preprocessing of the game (Sect. 4.3).

Basis strategies. We denote by $\lambda_1^{sec} \in \Lambda_1^{PM}(G)$ a pure memoryless worst-case winning strategy on G. This strategy exists in all states of the game due to the preprocessing, including states of the EC $U \in \mathcal{W}$. Still, it may require leaving the EC to ensure a strictly positive mean-payoff (because the definition of winning ECs only consider edges in E_Δ).

When using $\lambda_2^{\mathsf{stoch}}$, $\mathcal{P}_2$ cannot force leaving the winning EC $U \in \mathcal{W}$. By Thm. 2, $\mathcal{P}_1$ has a finite-memory strategy on $G_\Delta \restriction U$, denoted λ_1^{cmb}, that ensures eq. (2), and verifies eq. (1), *if we restrict* $\mathcal{P}_2$ to strategies in $\Lambda_2(G_\Delta)$.

Defining a witness-and-secure strategy. In order to prove Thm. 3, we define a pure finite-memory *witness-and-secure strategy* as follows.

Definition 5. *In a game G such that $P = G[\lambda_2^{\mathsf{stoch}}]$, $U \in \mathcal{W}$ is a winning EC and $s_{\mathsf{init}} \in U$ is the initial state, we define the* witness-and-secure strategy $\lambda_1^{wns} \in \Lambda_1^{PF}(G)$ *as follows.*

(i) Play the combined strategy $\lambda_1^{cmb} \in \Lambda_1^{PF}(G_\Delta \restriction U)$ as long as $\mathcal{P}_2$ picks edges in E_Δ.
(ii) As soon as $\mathcal{P}_2$ takes an edge in $E \setminus E_\Delta$,[13] play the worst-case winning strategy $\lambda_1^{sec} \in \Lambda_1^{PM}(G)$ forever.

This strategy makes use of the combined strategy λ_1^{cmb} presented in Def. 3. Hence it is similarly parameterized by two naturals K and L that respectively define the lengths of periods of type *(a)* and of type *(b)* in Def. 3. Again, we will show that for any $\varepsilon > 0$, we can find values for K and L such that Thm. 3 is verified. Notice that λ_1^{wns} is finite-memory since λ_1^{cmb} and λ_1^{sec} are too and watching for the appearance of an edge belonging to $E \setminus E_\Delta$ in the actions of the adversary only requires a finite amount of memory.

Intuitively, the witness-and-secure strategy acts as follows. As long as $\mathcal{P}_2$ conforms to E_Δ, playing in G is essentially the same as playing in G_Δ. Hence λ_1^{wns} prescribes acting like λ_1^{cmb}, which induces satisfaction of the BWC problem in $G_\Delta \restriction U$ by Thm. 2. Two requirements must be satisfied by λ_1^{wns}: (a) the worst-case and (b) the expected value. First consider (a). Two situations may occur. Either the outcome is such that λ_1^{wns} always stays in phase (i) and strategy λ_1^{cmb} is used forever in $G_\Delta \restriction U$, in which case direct application of Thm. 2 suffices to prove that eq. (1) is satisfied. Or, the outcome is such that λ_1^{wns} switches to phase (ii), in which case satisfaction of the worst-case requirement follows by definition of λ_1^{sec}. Now consider (b). Notice that the only consistent outcomes always stay in phase (i), by definition of the set $\mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1^{wns}, \lambda_2^{\mathsf{stoch}})$ which does not allow for choices outside of E_Δ. Hence the overall expectation is equal to the one over outcomes staying in phase (i). By Def. 5, the latter is exactly the expectation of λ_1^{cmb}, which satisfies the threshold by Thm. 2. Thus, the existence of fitting values of K and L for Thm. 3 is granted.

Illustration. Consider the winning EC U_2 in the game depicted in Fig. 2 and the initial state $s_6 \in U_2$. Notice that $\mathcal{P}_1$ can ensure a strictly positive mean-payoff in the subgame $G_\Delta \restriction U_2$, but not in $G \restriction U_2$. Indeed, it is easy to see that by always choosing the -1 edges (which requires edge $(s_7, s_6) \in E_\Delta \setminus E$), $\mathcal{P}_2$ can ensure a negative mean-payoff whatever the strategy of $\mathcal{P}_1$. However, there exists a strategy that ensures eq. (1), i.e., yields a strictly positive mean-payoff against any strategy in $\Lambda_2(G)$, by leaving the EC. Let λ_1^{sec} be the memoryless strategy that takes the edge (s_6, s_9) and then cycle on $(s_{10}s_9)^\omega$ forever: it guarantees a mean-payoff of $1 > 0$.

For a moment, consider the EC U_2 in G_Δ. Graphically, it means that the -1 edge from s_7 to s_6 disappears. In the subgame $G_\Delta \restriction U_2$, there are two particular memoryless strategies. The optimal worst-case strategy λ_1^{wc} guarantees a mean-payoff of $1/2 > 0$ by choosing to go to s_7. The optimal expectation strategy λ_1^{e} yields an expected mean-payoff of 3 by choosing to go to s_8 (notice this strategy yields the same expectation in $P_\Delta \restriction U_2$ and $P \restriction U_2$). Based on them, we build the combined strategy $\lambda_1^{cmb} \in \Lambda_1^{PF}(G_\Delta \restriction U_2)$ as defined in Def. 3 and by Thm. 2, for any $\varepsilon > 0$, there are values of K and L such that it satisfies the BWC problem for thresholds $(0, 3 - \varepsilon)$ in $G_\Delta \restriction U_2$. For example, for $K = L = 2$, we have $\mathbb{E}_{s_6}^{(P_\Delta \restriction U_2)[\lambda_1^{cmb}]}(\mathsf{MP}) = \mathbb{E}_{s_6}^{(P \restriction U_2)[\lambda_1^{cmb}]}(\mathsf{MP}) = 13/6$.

We construct the witness-and-secure strategy $\lambda_1^{wns} \in \Lambda_1^{PF}(G)$ based on λ_1^{cmb} and λ_1^{sec} as described by Def. 5. In this case, that means playing as λ_1^{cmb} until the -1 edge from s_7 to s_6 is taken by $\mathcal{P}_2$. As previously sketched, such a strategy ensures a worst-case mean-payoff equal to $1 > 0$ thanks to λ_1^{sec} and yields an expectation $\mathbb{E}_{s_6}^{P[\lambda_1^{wns}]}(\mathsf{MP}) = 13/6$ for $K = L = 2$.

[13] More complex switching schemes could be used, such as only switching if the edge taken is really dangerous (i.e., part of a non strictly positive cycle), switching after a bounded number of deviations from the support, etc. But this simple scheme proves to be sufficient to realize Thm. 3 and is easier to analyze.

Finally, notice that securing the mean-payoff by switching to phase (ii) of λ_1^{wns} *is needed* to satisfy the worst-case requirement if $\mathcal{P}_2$ plays in $E \setminus E_\Delta$. Also, remark that it is still necessary to alternate according to λ_1^{cmb} in $G_\Delta \restriction U_2$ and that playing λ_1^e is not sufficient to ensure the worst-case (because $\mathcal{P}_1$ has to deal with the -1 edge from s_8 to s_6 that remains in E_Δ).

Analysis of the witness-and-secure strategy. We close our discussion of winning ECs with the formal proof of Thm. 3 through the use of the witness-and-secure strategy λ_1^{wns} (Def. 5).

Proof (Theorem 3). Assume an arbitrary $\varepsilon > 0$. Let $K, L \in \mathbb{N}$ be such that the combined strategy $\lambda_1^{cmb} \in \Lambda_1^{PF}(G_\Delta \restriction U)$, as defined in Def. 3, satisfies the BWC problem for thresholds $(0, \nu^* - \varepsilon)$ in $G_\Delta \restriction U$. The existence of such values is granted by Thm. 2. We build the finite-memory strategy $\lambda_1^{wns} \in \Lambda_1^{PF}(G)$ according to Def. 5 and claim it satisfies the BWC problem for thresholds $(0, \nu^* - \varepsilon)$ in G.

First, consider the worst-case requirement. Let $\lambda_2 \in \Lambda_2(G)$ be any strategy of $\mathcal{P}_2$ and $\pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1^{wns}, \lambda_2)$ be any outcome consistent with λ_1^{wns}. Two cases are possible. One, $\mathcal{P}_2$ keeps choosing edges in E_Δ forever. That is, for $\pi = s_0 s_1 s_2 \ldots$, for all $i \geq 0$ such that $s_i \in S_\Delta = S_2$, we have that $(s_i, s_{i+1}) \in E_\Delta$. Then, the play is constrained to $G_\Delta \restriction U$ and consistent with λ_1^{cmb}. Hence, it follows from Thm. 2 (and Lemma 10) that $\mathsf{MP}(\pi) > 0$. Two, $\mathcal{P}_2$ chooses some edges in $E \setminus E_\Delta$. That is, for $\pi = s_0 s_1 s_2 \ldots$, there exists $i \geq 0$ such that $(s_i, s_{i+1}) \notin E_\Delta$. Let i_0 be the smallest index where it happens. By definition of λ_1^{wns}, we know that $\mathcal{P}_2$ switches to λ_1^{sec} at step i_0. Hence the suffix $\pi' = s_{i_0} s_{i_0+1} s_{i_0+2} \ldots$ is consistent with λ_1^{sec}. Consequently, $\mathsf{MP}(\pi') > 0$. By prefix-independence of the mean-payoff value function, we conclude that $\mathsf{MP}(\pi) > 0$, which closes the case of the worst-case requirement.

Second, consider the expected value requirement. We claim that $\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^{wns}, \lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) > \nu^* - \varepsilon$. By definition, $\mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1^{wns}, \lambda_2^{\mathsf{stoch}}) = \mathsf{Outs}_P(s_{\mathsf{init}}, \lambda_1^{wns})$ only contains plays where $\mathcal{P}_2$ conforms to E_Δ at all times. Such plays never exit the EC and by Def. 5, we have that $\mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1^{wns}, \lambda_2^{\mathsf{stoch}}) = \mathsf{Outs}_{G_\Delta}(s_{\mathsf{init}}, \lambda_1^{cmb}, \lambda_2^{\mathsf{stoch}})$. Also note that the probability measure of plays of those two sets is identical. Hence, we obtain

$$\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^{wns}, \lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) = \mathbb{E}_{s_{\mathsf{init}}}^{G_\Delta[\lambda_1^{cmb}, \lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) > \nu^* - \varepsilon$$

by Thm. 2 (and Lemma 11). This sets the case for the expectation and concludes our proof. □

4.8 Global strategy: favor reaching the highest valued winning end-components

We now have all the elements needed to describe the final steps of algorithm BWC_MP (lines 12-17) and prove its correctness. In this section, we first describe (Def. 6) how to modify the weights of the MDP $P = G[\lambda_2^{\mathsf{stoch}}]$ such that a classical optimal expectation strategy in the modified MDP P' will naturally tries to reach *winning* ECs with the highest combined expectation. This step is a cornerstone of the *global strategy* $\lambda_1^{glb} \in \Lambda_1^{PF}(G)$ that we define next (Def. 7). This strategy is a by-product of algorithm BWC_MP.

We study the adequacy of algorithm BWC_MP through two lemmas. In Lemma 12, we prove its *correctness*, i.e., that if it returns YES, then the global strategy λ_1^{glb} satisfies the BWC problem for the given thresholds. In Lemma 13, we show its *completeness*, i.e., that if it returns NO, then there exists no finite-memory strategy that satisfies the BWC problem. Combining those two lemmas and the analysis of the preprocessing conducted in Sect. 4.3, we conclude that algorithm BWC_MP is a valid algorithm to solve the BWC problem on any two-player game with the mean-payoff value function.

Modifying the MDP to naturally reach winning ECs. Our motivation is the creation of an MDP P' such that an optimal strategy in P' maximizes the expectation without using negligible states (as defined in Sect. 4.4, that is, with regard to G and P). Indeed, we know by Lemma 8 that winning ECs should be almost-surely eventually used in order to satisfy the worst-case requirement of the BWC problem. In particular, states in losing ECs and not in any winning sub-EC should be avoided in the long-run.

Definition 6. *Given $G = (\mathcal{G}, S_1, S_2)$, $\mathcal{G} = (S, E, w)$ and $P = G[\lambda_2^{\mathsf{stoch}}]$, we define $G' = (\mathcal{G}', S_1, S_2)$, $\mathcal{G}' = (S, E, w')$ and $P' = G'[\lambda_2^{\mathsf{stoch}}]$ by modifying the weight function as follows:*

$$\forall e = (s_1, s_2) \in E,\ w'(e) := \begin{cases} w(e) \text{ if } \exists U \in \mathcal{U}_{\mathrm{w}} \text{ s.t. } \{s_1, s_2\} \subseteq U, \\ 0 \text{ otherwise.} \end{cases}$$

Let $\lambda_1^e \in \Lambda_1^{PM}(P')$ be a pure memoryless strategy of $\mathcal{P}_1$ that maximizes the expected mean-payoff in P'. Such a strategy always exists [14]. Note that following λ_1^e does not suffice to satisfy the BWC problem in general (Rem. 6). This strategy will be part of the global strategy λ_1^{glb} (Def. 7): its role is to maximize the combined expectation of reachable winning ECs, while avoiding using negligible states, in particular states that only belong to losing ECs. The strategy to adopt inside winning ECs will be prescribed by another part of the global strategy, based on what we have established in Sect. 4.7. Observe that it suffices to consider the *maximal* winning ECs in order to maximize the expectation, as proved by Lemma 6.

Remark 8. Notice that λ_1^e is also well-defined in P and G thanks to the shared underlying graph. Also, recall that all states of G are worst-case winning due to the preprocessing. Let $\lambda_1^{wc} \in \Lambda_1^{PM}(G)$ be an optimal worst-case winning strategy. We observe that all states remain worst-case winning in G' for the reason that an optimal worst-case strategy only needs to visit edges involving negligible states finitely often. Indeed, either these negligible states do not belong to any EC, in which case $\mathcal{P}_1$ cannot rely on them to satisfy the worst-case requirement (as he cannot ensure that he will be able to see them infinitely often), or they belong to losing ECs and no winning sub-EC, in which case $\mathcal{P}_1$ only needs to visit them a finite number of times (basically to get out of the set $(\bigcup_{U\in\mathcal{E}} U) \setminus (\bigcup_{U\in\mathcal{W}} U)$ if the play starts in it and reach the set $\bigcup_{U\in\mathcal{W}} U$). Hence, the guaranteed mean-payoff is not impacted by the changes described in Def. 6: it remains strictly positive in all states. By virtue of this, we deduce that

$$\forall s \in S,\ \mathbb{E}_{s_{\mathsf{init}}}^{P'[\lambda_1^e]}(\mathsf{MP}) \geq \mathbb{E}_{s_{\mathsf{init}}}^{P'[\lambda_1^{wc}]}(\mathsf{MP}) > 0,$$

and as such, that strategy λ_1^e will not prescribe staying in the set $(\bigcup_{U\in\mathcal{E}} U) \setminus (\bigcup_{U\in\mathcal{W}} U)$ forever. Indeed, it is always beneficial to exit it and obtain a strictly positive expectation instead of an expectation equal to zero (recall all edges involving negligible states are mapped to weight zero by Def. 6).

Defining a global strategy. Based on the memoryless strategies λ_1^e and λ_1^{wc} in G (as defined above), and the pure finite-memory witness-and-secure strategy λ_1^{wns} in winning ECs (as presented in Def. 5),[14] we build a *global strategy* λ_1^{glb} in G as follows. This strategy is parameterized by a natural constant $N \in \mathbb{N}$.

Definition 7. *In a game G, we define the* global strategy $\lambda_1^{glb} \in \Lambda_1^{PF}(G)$ *as follows.*

(a) *Play $\lambda_1^e \in \Lambda_1^{PM}(G)$ for N steps.*
(b) *Let $s \in S$ be the reached state.*
  ($b.1$) *If $s \in U \in \mathcal{U}_{\mathrm{w}}$, play the corresponding strategy $\lambda_1^{wns} \in \Lambda_1^{PF}(G)$ forever.*
  ($b.2$) *Else play $\lambda_1^{wc} \in \Lambda_1^{PM}(G)$ forever.*

Let us sketch this strategy. In phase (a), the optimal expectation strategy in P' is followed. It will drive the outcomes towards the ECs with the highest expected values. By taking N large enough, we can ensure that the probability of being in an EC will be arbitrarily close to one (by Lemma 5). As a result of the weights modification described in Def. 6, we can further ensure that the probability of being inside a *winning* EC will be arbitrarily close to one. Remark that in phase $(b.1)$, the witness-and-secure strategy guarantees satisfaction of the worst-case requirement while granting an expectation arbitrarily close to the optimal expectation of the EC (as proved in Sect. 4.7). Also, in phase $(b.2)$, the mean-payoff of outcomes is strictly positive, and the probability of being in $(b.2)$ can be arbitrarily close to zero for large enough values of N. Overall, we obtain that λ_1^{glb} satisfies the worst-case requirement because the strategies played in the two terminal phases, $(b.1)$ and $(b.2)$, all guarantee its satisfaction and the mean-payoff is prefix-independent (hence it is not impacted by phase (a)). Furthermore, the expectation of λ_1^{glb} can be arbitrarily close to the maximal expectation v^* achievable in P' (i.e., the one achieved by λ_1^e) by taking sufficiently large values for the constants K, L and N. Hence, if $v^* > v$, λ_1^{glb} is a proper BWC satisfying strategy for $\mathcal{P}_1$.

Finally, v^* constitutes an upper bound to the expectation of any strategy of $\mathcal{P}_1$ in P'. By Lemma 8, it is also an upper bound on the expectation of any strategy that satisfies the worst-case requirement in the original game and MDP. It follows that if $v^* \leq v$, then there exists no finite-memory strategy that satisfies the BWC problem.

[14] Parameters K and L may vary depending on the actual corresponding EC.

As the validity of the preprocessing was shown in Sect. 4.3, this let us conclude that algorithm BWC_MP is both correct *and* complete.

Illustration. Consider the game G depicted in Fig. 2 and the associated MDP $P = G[\lambda_2^{\mathsf{stoch}}]$. Following Lemma 5, analysis of the maximal ECs U_1, U_2 and U_3 reveals that the maximal expected mean-payoff achievable in P is 4. It is for example obtained by the memoryless strategy that chooses to go to s_2 from s_1 and to s_4 from s_3. Remark that playing in U_1 forever is needed to achieve this expectation. By Lemma 8, this should not be allowed as the worst-case cannot be ensured if it is. Indeed, $\mathcal{P}_2$ can produce worst-case losing outcomes by playing the -1 edge. Clearly, the maximal expected value that $\mathcal{P}_1$ can ensure while guaranteeing the worst-case requirement is thus bounded by the maximal expectation in P', i.e., by 3. Let λ_1^e denote an optimal memoryless expectation strategy in P' that tries to enter U_2 by playing (s_1, s_2) and (s_3, s_5), and then plays edge (s_6, s_8) forever.

Observe that algorithm BWC_MP answers YES for any thresholds pair $(0, \nu)$ such that $\nu < 3$. For the sake of illustration, we construct the global strategy λ_1^{glb} as presented in Def. 7 with $N = 6$ and $K = L = 2$. For the first six steps, it behaves exactly as λ_1^e. Note that after the six steps, the probability of being in U_2 is $1/4 + 1/8 = 3/8$. Then, λ_1^{glb} switches to another strategy depending on the current state (λ_1^{wns} or λ_1^{wc}) and sticks to this strategy forever. Particularly, if the current state belongs to U_2, it switches to λ_1^{wns} as described in Def. 5 for $K = L = 2$, which guarantees the worst-case threshold and induces an expectation of $13/6$ (Sect. 4.7). By definition of λ_1^{glb} on the sample game G, if the current state after six steps is not in U_2, then λ_1^{glb} switches to λ_1^{wc} which guarantees a mean-payoff of 1 by reaching state s_9 and then playing $(s_9 s_{10})^\omega$. Overall, the expected mean-payoff of λ_1^{glb} against $\lambda_2^{\mathsf{stoch}}$ is

$$\mathbb{E}_{s_1}^{G[\lambda_1^{glb}, \lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) \geq \frac{3}{8} \cdot \frac{13}{6} + \frac{5}{8} \cdot 1 = \frac{23}{16}.$$

Notice that by taking N, K and L large enough, it is possible to satisfy the BWC problem for any $\nu < 3$ with the strategy λ_1^{glb}. Also, observe that the winning EC U_2 is crucial to achieve expectations strictly greater than 2, which is the upper bound when limited to EC U_3. For example, $N = 25$ and $K = L = 2$ implies an expectation strictly greater than 2 for the global strategy.

Lastly, note that in general, the maximal expectation achievable in P' (and thus in P when limited to strategies that respect the worst-case requirement) may depend on a combination of ECs instead of a unique one. This is transparent through the solving of the expected value problem in the MDP P'. Hence, the approach followed by algorithm BWC_MP is a way of solving a complex problem by breaking it into smaller pieces.

Correctness and completeness. We start by proving the correctness of the algorithm BWC_MP described in Alg. 1 and the soundness of the global strategy presented in Def. 7 to satisfy the BWC problem.

Lemma 12 (correctness). *If algorithm* BWC_MP *answers* YES*, then there exist values of the parameters such that the global strategy $\lambda_1^{glb} \in \Lambda_1^{PF}$ satisfies the BWC mean-payoff problem.*

Proof. We assume the answer returned by BWC_MP is YES and we prove the claim.

First, consider the worst-case requirement (eq. (1)). Let $\lambda_2 \in \Lambda_2$ be an arbitrary strategy of $\mathcal{P}_2$. Let N take an arbitrary value in $\mathbb{N}$, and for any winning EC $U \in \mathcal{U}_{\mathrm{w}}$, let K_U take an arbitrary value in $\mathbb{N}$ and L_U be defined according to Def. 4 with regard to K_U. Consider the outcomes consistent with λ_1^{glb} and λ_2. Our goal is to prove that for all outcomes $\pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1^{glb}, \lambda_2)$, we have that $\mathsf{MP}(\pi) > 0$. Let π be an arbitrary outcome in this set, $s = \mathsf{Last}(\pi(N))$ be the state reached after phase (a) of the global strategy, and π' be the suffix play such that $\pi = \pi(N) \cdot \pi'$. Two cases are possible. First, assume $s \in U$ for some maximal winning EC $U \in \mathcal{U}_{\mathrm{w}}$. Then, π' is consistent with the witness-and-secure strategy λ_1^{wns} (as presented in Sect. 4.7 for initial states in U). By Thm. 3, $\mathsf{MP}(\pi') > 0$. Second, assume $s \notin \bigcup_{U \in \mathcal{U}_{\mathrm{w}}} U$, i.e., $s \in S_{\mathsf{neg}}$. Then, π' is consistent with the worst-case winning strategy λ_1^{wc} provided by the preprocessing, and we have that $\mathsf{MP}(\pi') > 0$. By prefix-independence of the mean-payoff value function, we conclude that in both cases, $\mathsf{MP}(\pi) = \mathsf{MP}(\pi') > 0$, proving that strategy λ_1^{glb} ensures the worst-case requirement.

Second, consider the expected value requirement (eq. (2)). We need to prove that $\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^{glb}, \lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) > \nu$ for some well-chosen values of N and $K \in \mathbb{N}$. Formally, the value K_U may be different in each winning EC $U \in \mathcal{U}_{\mathrm{w}}$, so we will take a uniform value K sufficiently large to ensure that it works for all ECs. Values $L_U(K)$ are defined according

to Def. 4. Again noting that the weights encountered during phase (a) of strategy λ_1^{glb} have no impact on the mean-payoff of plays (because phase (a) is of finite duration and all weights are also finite), we formulate the expectation as

$$\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^{glb},\lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) = \sum_{U\in\mathcal{U}_{\mathrm{W}}} \big[p_N(U)\cdot e_K(U)\big] + \sum_{s\in S_{\mathsf{neg}}} \big[p_N(s)\cdot e_{\mathrm{WC}}(s)\big], \tag{12}$$

where $p_N(U)$ denotes the probability to be in a state belonging to the maximal winning EC $U \in \mathcal{U}_{\mathrm{W}}$ after N steps of following strategy λ_1^e (i.e., phase (a)); $e_K(U)$ denotes the expectation of plays starting in U and consistent with λ_1^{wns} for values K and $L_U(K)$ of the parameters (this expectation is identical for all initial states in the EC); $p_N(s)$ denotes the probability to be in a given negligible state $s \in S_{\mathsf{neg}}$ (i.e., outside of winning ECs) after phase (a); and $e_{\mathrm{WC}}(s)$ denotes the expectation over plays that start in such a state s and are consistent with the worst-case strategy λ_1^{wc}. Observe that $\sum_{s\in S_{\mathsf{neg}}} p_N(s) = 1 - \sum_{U\in\mathcal{U}_{\mathrm{W}}} p_N(U)$.

Similarly, we write the expectation of the optimal expectation strategy λ_1^e in P' as

$$\mathbb{E}_{s_{\mathsf{init}}}^{G'[\lambda_1^{e},\lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) = \sum_{U\in\mathcal{U}_{\mathrm{W}}} \big[p(U)\cdot e(U)\big], \tag{13}$$

where $p(U)$ and $e(U)$ denote the probability and the expectation of maximal winning ECs when strategy λ_1^e is followed forever. Note that eq. (13) depends uniquely on winning ECs by consequence of Rem. 8, and specifically maximal winning ECs by further application of Lemma 6. In addition, observe that

$$\mathbb{E}_{s_{\mathsf{init}}}^{G'[\lambda_1^{e},\lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) = \mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^{e},\lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) = \nu^*,$$

since the weight modification of Def. 6 does not alter winning ECs.

We claim that $\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^{glb},\lambda_2^{\mathsf{stoch}}]}(\mathsf{MP})$ tends to ν^* when N and K tend to infinity. We study the terms of eq. (12). Note that e_{WC} takes a bounded value (in $]0, W]$ by definition of λ_1^{wc}). By application of the analysis developed in Lemma 11 and Theorem 3, we have that

$$\forall U \in \mathcal{U}_{\mathrm{W}},\ e_K(U) \xrightarrow[K\to\infty]{} e(U).$$

Furthermore, by definition of λ_1^e we have that

$$\forall U \in \mathcal{U}_{\mathrm{W}},\ p_N(U) \xrightarrow[N\to\infty]{} p(U),$$

and by definition of the modified weight function (Def. 6) and Rem. 8, that

$$\sum_{U\in\mathcal{U}_{\mathrm{W}}} p_N(U) \xrightarrow[N\to\infty]{} \sum_{U\in\mathcal{U}_{\mathrm{W}}} p(U) = 1.$$

Summing up, we obtain that $\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^{glb},\lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) \xrightarrow[N,K\to\infty]{} \nu^*$. By convergence, for all $\varepsilon > 0$, there exist $N, K \in \mathbb{N}$ such that $\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^{glb},\lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) \geq \nu^* - \varepsilon$. Since algorithm BWC_MP answered YES, we have that $\nu^* > \nu$. Hence, there exist values $N, K \in \mathbb{N}$ such that $\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^{glb},\lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) > \nu$. This concludes the proof. □

In order to prove that the algorithm solves the BWC problem (Def. 1) for the mean-payoff value function, we still need to establish its completeness: if the global strategy does not suffice to satisfy some thresholds pair, then no finite-memory strategy can do it.

Lemma 13 (completeness). *If algorithm* BWC_MP *answers* NO*, then there exists no finite-memory strategy of* $\mathcal{P}_1$ *that satisfies the BWC mean-payoff problem.*

Proof. By contradiction, assume there exists $\lambda_1^f \in \Lambda_1^F$ that satisfies the BWC problem for thresholds $(0, \nu)$. We claim that algorithm BWC_MP answers YES. First, notice that the algorithm cannot answer NO at line 5 since λ_1^f satisfies

the worst-case requirement from the initial state s_{init}. Hence it remains to prove that ν^*, as computed by the algorithm, is such that $\nu^* > \nu$. If it is the case, the algorithm will answer YES, which proves our claim.

By hypothesis, strategy λ_1^f induces an expectation $\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^f,\lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) > \nu$. By Lemma 8, we have that

$$\mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1^f,\lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) = \mathbb{E}_{s_{\mathsf{init}}}^{G'[\lambda_1^f,\lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) = \mathbb{E}_{s_{\mathsf{init}}}^{P'[\lambda_1^f]}(\mathsf{MP}),$$

with G' the game obtained by the transformation defined in Def. 6. Moreover, by definition of the optimal expectation, we have that for all $\lambda_1 \in \Lambda_1$, $\nu^* \geq \mathbb{E}_{s_{\mathsf{init}}}^{P'[\lambda_1]}(\mathsf{MP})$. In particular, this inequality is verified for strategy λ_1^f. Hence, we obtain that $\nu^* \geq \mathbb{E}_{s_{\mathsf{init}}}^{P'[\lambda_1^f]}(\mathsf{MP}) > \nu$. Consequently, the answer of the algorithm is YES and the lemma is proved. □

In summary, correctness and completeness of algorithm BWC_MP as stated in Thm. 1 follows from the combination of Lemma 12, Lemma 13 and the validity of the preprocessing, as presented in Sect. 4.3. The complexity of the algorithm is discussed in the next section (Lemma 14), as well as matching lower bounds for the BWC problem (Lemma 15).

4.9 Complexity: algorithm and lower bound

In this section, we prove that algorithm BWC_MP is in NP ∩ coNP (Lemma 14). The classical worst-case threshold problem also belongs to NP ∩ coNP [30,23] and whether it is in P or not is a long-standing open problem [3,6]. In Lemma 15, we establish that it reduces in polynomial time to the BWC problem. Given the outstanding nature of the worst-case threshold problem membership to P, algorithm BWC_MP can thus be considered optimal. Furthermore, we observe that if the worst-case threshold problem were proved to be solvable in deterministic polynomial time, then algorithm BWC_MP would also be in P (Rem. 9).

To prove the NP ∩ coNP membership of the algorithm, we study each of its computing steps and observe that they all require polynomial time in the size of the input, except for a polynomial number of calls to an NP ∩ coNP algorithm solving the worst-case threshold problem. Rem. 9 is also induced by this observation.

Lemma 14. *Algorithm* BWC_MP *is in NP ∩ coNP.*

Proof. To begin with, remark that the size of the *input* depends polynomially on (i) the number of states of the input game $|S^i|$, (ii) the number of edges of the input game $|E^i|$, (iii) the number of bits of the encoding of weights $V^i = \log_2 W^i$, (iv) the size of the memory of the Moore machine $|\mathsf{Mem}|$, (v) the size of the supports and the length of the encoding of probabilities for the next-action function α_{n}, and (vi) the encoding of the thresholds $\mu, \nu \in \mathbb{Q}$.

To prove the NP ∩ coNP membership of the algorithm, we review each step sequentially. Lines 1-2 and 4-10 are at most polynomial in the input. Line 3 consists in solving the worst-case threshold problem on the input game G^i: this can be done by calling an NP ∩ coNP algorithm [30,23]. Overall, the preprocessing is in NP ∩ coNP and produces a game G such that $|G| \leq |G^i| \cdot |\mathcal{M}(\lambda_2^i)|$, using the natural definitions of those sizes as polynomial functions of the values described in points (i) to (vi).

For the main algorithm, the complexities are as follows. Line 11 is the call to the sub-algorithm MWEC(P_Δ), which has been proved to work in NP ∩ coNP in Lemma 7. Note that the size of $P = G[\lambda_2^{\mathsf{stoch}}]$ is polynomial in the input. The weights modification (line 12) requires linear time (polynomial in the input game) as do lines 14-17. Finally, computing the maximal expected value on P' (line 13) is polynomial in $|P'|$ via linear programming [14], hence polynomial in the input size.

In conclusion, we observe that all operations of the algorithm are executed at most once, and each of them belongs to NP ∩ coNP, which proves the claim. □

Remark 9. Assume an algorithm PTIME_WC is established to solve the worst-case threshold problem in deterministic polynomial time. Then, the complexities of algorithm BWC_MP and sub-algorithm MWEC boil down to a polynomial number of polynomial time operations and external calls, and it follows that BWC_MP is in P.

Reduction of the worst-case threshold problem to the BWC one seems natural by eq. (1). Still, we need to pay attention to the strict inequality in the BWC problem definition: we use the existence of memoryless winning strategies for the worst-case problem and careful analysis of the domain of the mean-payoff values of outcomes to prove that it is not restrictive. The expected value part of the BWC problem can be defined arbitrarily under certain conditions.

Lemma 15. *The worst-case threshold problem on mean-payoff games reduces in polynomial time to the BWC mean-payoff problem.*

Proof. Given a game $G = (\mathcal{G}, S_1, S_2)$, its underlying graph $\mathcal{G} = (S, E, w)$, an initial state $s_{\mathsf{init}} \in S$, and the worst-case threshold $\mu = 0$ (without loss of generality), the worst-case threshold problem asks if the following proposition is true:

$$\exists \lambda_1 \in \Lambda_1, \forall \lambda_2 \in \Lambda_2, \forall \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2), \mathsf{MP}(\pi) \geq 0. \tag{14}$$

By the results of [24,13], it is equivalent to restrict both players to memoryless strategies:

$$\exists \lambda_1^{pm} \in \Lambda_1^{PM}, \forall \lambda_2^{pm} \in \Lambda_2^{PM}, \forall \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1^{pm}, \lambda_2^{pm}), \mathsf{MP}(\pi) \geq 0. \tag{15}$$

It is well-known that in this context, $\mathsf{MP}(\pi) \geq 0 \Leftrightarrow \mathsf{MP}(\pi) > -\frac{1}{|S|}$. Indeed, consider the following argument. First, the mean-payoff of any outcome $\pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1^{pm}, \lambda_2^{pm})$ can be trivially bounded by $-W \leq \mathsf{MP}(\pi) \leq W$, with W the largest absolute value of any weight assigned by w to edges of G. Second, consider the decomposition of π into simple cycles (i.e., cycles with no repeated state except for the starting and ending state). Since weights are integers, any simple cycle has an associated mean-payoff belonging to $\{-W, \ldots, -\frac{1}{|S|}, 0, \frac{1}{|S|}, \ldots, W\}$. As both strategies are memoryless, any outcome $\pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1^{pm}, \lambda_2^{pm})$ will ultimately consist in a repeated simple cycle. Hence we have that $\mathsf{MP}(\pi) \in \{-W, \ldots, -\frac{1}{|S|}, 0, \frac{1}{|S|}, \ldots, W\}$ and we observe that no value can be taken between $-\frac{1}{|S|}$ and 0.

Consequently, eq. (15) is equivalent to

$$\exists \lambda_1^{pm} \in \Lambda_1^{PM}, \forall \lambda_2^{pm} \in \Lambda_2^{PM}, \forall \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1^{pm}, \lambda_2^{pm}), \mathsf{MP}(\pi) > -\frac{1}{|S|}. \tag{16}$$

To formulate eq. (16) in terms of a BWC problem, we have to define an expected value threshold $\nu \in \mathbb{Q}$ and a stochastic model $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^F$. Since all plays $\pi \in \mathsf{Plays}(\mathcal{G})$ satisfy $\mathsf{MP}(\pi) \geq -W$ by definition of the weight function, we trivially have that

$$\forall \lambda_1 \in \Lambda_1, \forall \lambda_2^{\mathsf{stoch}} \in \Lambda_2^F, \mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1, \lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) \geq -W.$$

Hence, it suffices to fix an arbitrary stochastic model $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^F$ and an arbitrary expectation threshold $\nu < -W$ to obtain that eq. (14) is satisfied *if and only if* $\mathcal{P}_1$ has a strategy to satisfy the BWC problem for thresholds $(-\frac{1}{|S|}, \nu)$ against the stochastic model $\lambda_2^{\mathsf{stoch}}$.

Notice this reduction is polynomial because we can choose a simple stochastic model (e.g., a memoryless strategy requires a Moore machine of size linear in the size of the game) and a value of ν that will not require a super-polynomial growth of the encodings (e.g., $\nu = -W - 1$). □

Our complexity results are summed up in Thm. 1.

4.10 Memory requirements

Across the previous sections, we have studied the complexity of deciding the BWC problem, i.e., deciding the existence of a *finite-memory* strategy of $\mathcal{P}_1$ satisfying Def. 1 for the mean-payoff value function. Now, we focus on the *size of the memory* used by such a strategy. In Thm. 4, we give an upper bound for the memory of the global strategy described in Def. 7 (which has been shown to suffice if satisfaction of the BWC problem is possible). This is obtained through careful analysis of the structure of involved strategies (global, witness-and-secure, combined). All of them are based on alternation between well-chosen pure memoryless strategies, based on parameters N, K and $L \in \mathbb{N}$. We prove that these values only need to be polynomial in the size of the game and the stochastic model, and in the values of weights and thresholds, granting the claim.

Furthermore, we prove this upper bound to be tight in the sense that polynomial memory in the values of weights is needed in general. To establish this result, we provide a family of games $(G(X))_{X \in \mathbb{N}_0}$, reduced to a winning EC and where all possible edges are assigned non-zero probability by the stochastic model (i.e., verifying Assumption 1). This family is presented in Fig. 5. By choosing the worst-case threshold to be $\mu = 0$ and the expectation threshold to be $\nu \in]1, 5/4[$, we ensure that the BWC problem is satisfiable and that it cannot be achieved by the memoryless strategy

that always chooses edge (s_1, s_2). Intuitively, it is thus mandatory to choose (s_1, s_3) infinitely often in order to achieve the BWC problem. Moreover, after some point, everytime this edge is chosen, a satisfying strategy must be able to *eventually* counteract the potential negative weight $-X$ by taking edge (s_1, s_2) for $\lfloor X/2 \rfloor + 1$ times. This proves that polynomial memory in W is needed.

Theorem 4. *Memory of pseudo-polynomial size may be necessary and is always sufficient to satisfy the BWC problem for the mean-payoff: polynomial in the size of the game and the stochastic model, and polynomial in the weight and threshold values.*

Proof. We first consider the upper bound on memory, derived by analysis of the global strategy λ_1^{glb} (Def. 7). Observe that it follows the memoryless strategy λ_1^e for N steps before switching to phase *(b)*. The correctness of the strategy (Lemma 12) relies on the existence of a value N such that the probability of being in an EC after N steps is *high enough*. We argue that N does not need to be exponentially large.

Consider the probability to be outside of winning ECs after N steps. Applying classical results on Markov chains, we obtain that this probability decreases exponentially fast when N grows. Indeed, to prove it, it suffices to consider the chain $G[\lambda_1^e, \lambda_2^{\mathsf{stoch}}]$, replace BSCCs by absorbing states and observe that the probability of absorption tends towards one exponentially fast [20]. Now, consider the expectation of the global strategy for given constants N and K, as given in eq. (12). Let $\nu < \nu^* - \varepsilon$ be the expected value threshold considered in the BWC problem (as before, we assume $\mu < \nu < \nu^*$ otherwise the problem is trivial). Assume that K is sufficiently large to have $\sum_{U \in \mathcal{U}_w} p(U) \cdot e_K(U) > \nu^* - \varepsilon'$, with $\varepsilon' < \varepsilon$. We want to establish how large N needs to be to ensure an overall expectation strictly greater than ν. Since $e_{\mathrm{WC}}(s)$ can be trivially lower bounded by zero for all $s \in S_{\mathsf{neg}}$, it is clear that to obtain the needed property, we need to have values $p_N(U)$ growing polynomially with ε and ν^*. As the growth of $p_N(U)$ is exponential in the growth of the value N, we obtain that a logarithmic value of N, hence *polynomial in the encoding*, suffices to achieve the desired expected value.

Similarly, we study the strategies followed in phase *(b)* of the global strategy (Def. 7). The case *(b.2)* is the easiest: the worst-case strategy λ_1^{wc} is memoryless. In case *(b.1)*, the witness-and-secure strategy λ_1^{wns} is used. By Def. 5, this strategy needs polynomial memory to witness the use of edges in $E \setminus E_\Delta$ and to implement the memoryless secure strategy λ_1^{sec}. It also needs to implement the combined strategy λ_1^{cmb}, based on alternation between memoryless strategies. The size of the memory of λ_1^{cmb} is polynomial in K, L and the largest absolute value taken by Sum, as well as in the size of the game. The proof of Lemma 9 guarantees that for constant K, a value polynomial in the size of the input game and the stochastic model, as well as in the values of weights and thresholds, suffices. By Def. 4, an identical situation is verified for L. Finally, the running sum Sum takes values in $\{-K \cdot W, \ldots, K \cdot W\}$, hence it also verifies such bounds. Overall, the memory needed by the combined strategy is polynomial in the size of the input game and the stochastic model, and in the weight and threshold values.

Aggregating all these bounds, we conclude that the global strategy also requires memory at most polynomial in the size of the input game and the stochastic model, and in the values, thus proving the upper bound.

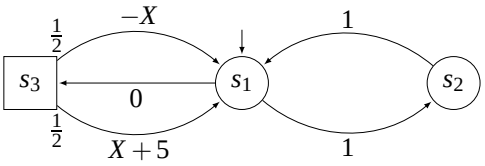

Fig. 5: Family of games $(G(X))_{X \in \mathbb{N}_0}$ requiring polynomial memory in $W = X + 5$ to satisfy the BWC problem for thresholds $(0, \nu \in]1, 5/4[)$.

It remains to show that pseudo-polynomial memory is really necessary in general. In order to achieve this, we introduce a family of games, $(G(X))_{X \in \mathbb{N}_0}$, such that winning the BWC problem on $G(X)$ requires memory polynomial in the largest weight $W = X + 5$. This family is presented in Fig. 5. Let the worst-case threshold be $\mu = 0$ and the expectation threshold be an arbitrary value $\nu \in]1, 5/4[$. Thanks to Thm. 2, the BWC problem is satisfiable, because $G(X)$ is reduced to a winning EC with no edge of probability zero, and the optimal expectation is $5/4 > \nu$ (expectation

achieved by the memoryless strategy that always chooses edge (s_1, s_3)). Notice that it cannot be achieved by the memoryless strategy that always chooses edge (s_1, s_2) since this strategy induces a mean-payoff equal to $1 < \nu$. Hence it is mandatory to choose (s_1, s_3) infinitely often in order to achieve the expected value requirement (eq. (2)).

Let $\lambda_1 \in \Lambda_1^F$ be a finite-memory strategy of $\mathcal{P}_1$ that satisfies the BWC problem. Observe that it may as well be pure, i.e., $\lambda_1 \in \Lambda_1^{PF}$ as choosing edge (s_1, s_3) with a non-zero probability recurrently yields consistent outcomes that do not satisfy the worst-case requirement (eq. (1)). Also observe that anytime edge (s_1, s_3) is chosen, there is a probability $1/2$ that the edge of weight $-X$ is taken to come back. Hence, from some point on, every appearence of this edge of weight $-X$ must be *eventually* counteracted in order to preserve the worst-case requirement. A finite number of non-compensated occurences is not a problem thanks to the prefix-independence of the mean-payoff value function. Looking at the involved weights, it is clear that taking the edge (s_1, s_2) for $(\lfloor X/2 \rfloor + 1)$ times is necessary to counteract the negative edge of weight $-X$. Hence, memory polynomial in X (hence in W) is needed to ensure both the worst-case and the expected value requirements for the given thresholds. This concludes our proof. □

4.11 Infinite-memory strategies

We close our study of the BWC problem for the mean-payoff value function by considering what happens when $\mathcal{P}_1$ is allowed to use infinite-memory strategies. Specifically, we show that in this context, infinite-memory strategies are in general strictly more powerful than finite-memory ones: they can exploit losing ECs to benefit from their possibly higher optimal expected value; and even inside a single winning EC, they can be optimal with regard to the expectation whereas finite-memory ones are limited to ε-optimality. Nonetheless, as discussed in the introduction, such strategies are ill-suited for the synthesis of implementable controllers for real-world applications, hence our focus on finite memory in the previous results.

Losing end-components may still be useful. Let us consider the game depicted in Fig. 6, together with a memoryless stochastic model of the adversary $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^M$, modeled by the probabilities $1/10$ and $9/10$ on the edges leaving s_1. The MDP $P = G[\lambda_2^{\mathsf{stoch}}]$ can be decomposed into two end-components U_1 and U_2, as depicted by the dashed lines. Assume the worst-case threshold is $\mu = -3/2$ (notice for once we take it different than zero), then U_1 is losing (because $\mathcal{P}_2$ can induce an outcome of mean-payoff value $-4/2 \leq -3/2$, and he can do that by choosing edges in E_Δ) and U_2 is winning (as the only outcome yields mean-payoff $-1 > -3/2$), following Def. 2.

As shown in Lemma 8, any finite-memory strategy of $\mathcal{P}_1$ which ensures a mean-payoff strictly greater than μ, leaves U_1 with probability one against $\lambda_2^{\mathsf{stoch}}$, because states of U_1 are classified as negligible. Therefore, in order to satisfy the worst-case requirement of the BWC problem, the expected mean-payoff of any finite-memory strategy of $\mathcal{P}_1$ is $\nu_2^* = -1$, i.e., the expectation obtained in U_2 by the only possible outcome. Notice however that if we forget about the worst-case requirement, the maximal expectation that $\mathcal{P}_1$ could achieve in U_1 is $\nu_1^* = \frac{1}{2} \cdot (\frac{9}{10} \cdot 4 + \frac{1}{10} \cdot (-4)) = \frac{8}{5}$.

We now show that $\mathcal{P}_1$ can ensure the worst-case requirement and obtain an expected value strictly greater than -1, if he is allowed to use infinite memory. We define a pure infinite-memory strategy λ_1 for $\mathcal{P}_1$ as follows: λ_1 stores the running sum along the prefix played so far, and chooses to move from s_0 to s_1 as long as this sum is strictly greater than zero (except in the first round where it moves directly to s_1). First, notice that this strategy trivially guarantees a mean-payoff greater than or equal to $-1 > \mu$. Indeed, either the running sum always stays strictly positive, implying that the mean-payoff is at least zero, or the running sum gets negative or null at some point, in which case the strategy switches to U_2 and the mean-payoff takes value -1. Second, let us compute the expected mean-payoff of this strategy against $\lambda_2^{\mathsf{stoch}}$. Let p_{switch} denote the probability to switch to U_2 along a play, as prescribed by the strategy λ_1. By definition, it is equal to the probability, when playing inside the EC U_1 and starting from an initial credit of zero in state s_0, to come back to s_0 with a credit less than or equal to zero after an arbitrary number of steps. Formally, let M be the MC induced by the subgame $G \restriction U_1$ and $\lambda_2^{\mathsf{stoch}}$ (note that $\mathcal{P}_1$ has no choice in it). We have

$$p_{\mathsf{switch}} = \mathbb{P}_{s_0}^M\big(\{\pi \in \mathsf{Outs}_M(s_0) \mid \exists i > 0, \mathsf{Last}(\pi(i)) = s_0 \wedge \mathsf{TP}(\pi(i)) \leq 0\}\big).$$

Determining the probability p_{switch} of the running sum hitting zero is equivalent to a well-studied problem on Markov chains, known as the *gambler's ruin problem* (see for instance [20]). Applying results on this problem, we obtain that, if the probabilities $9/10$ and $1/10$ are respectively replaced by arbitrary probabilities p and q such that $p > q$, the probability that the gambler is eventually ruined is $\frac{q}{p}$. In our example, this implies that $p_{\mathsf{switch}} = 1/9$.

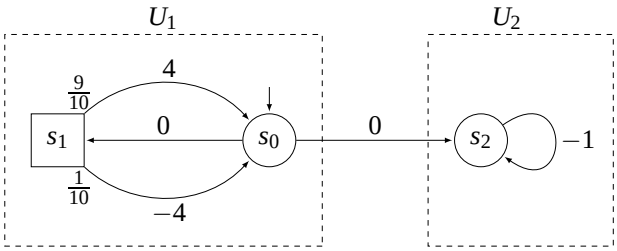

Fig. 6: Infinite-memory strategies may use losing ECs forever with a non-zero probability in order to increase the expected value.

We are now able to use this result to provide a lower bound for the expected value of the strategy λ_1. Consider the set of outcomes $\mathsf{Outs}_G(s_0, \lambda_1, \lambda_2^{\mathsf{stoch}})$: it can be partitioned into the set of plays that stay in U_1, for which the mean-payoff is trivially bounded by zero as discussed before; and the set of plays that reach U_2, for which the mean-payoff is equal to -1. Hence, the overall expectation respects the following inequality:

$$\mathbb{E}_{s_0}^{G[\lambda_1, \lambda_2^{\mathsf{stoch}}]}(\mathsf{MP}) \geq p_{\mathsf{switch}} \cdot (-1) + (1 - p_{\mathsf{switch}}) \cdot 0 = -\frac{1}{9} > -1.$$

Clearly, we see that strategy λ_1 yields an expectation at least equal to $-1/9$, hence strictly greater than the expectation achievable by any finite-memory strategy satisfying the BWC problem, which we have shown to be equal to -1.

Intuitively, the added power given by infinite memory comes from the possibility to memorize an unbounded running sum of weights, whereas finite memory implies an upper bound on such a sum. In the first case, $\mathcal{P}_1$ will be able to properly acknowledge that some plays see their running sum diverging without ever dropping to zero (the set of such plays has a strictly positive probability in our example), which lets him benefit from the added expectation without endangering the worst-case requirement. In the second case, $\mathcal{P}_1$ sees all running sums as upper bounded by some value $X \in \mathbb{N}$ due to its limited memory. As such, when he sees a sequence of weights whose total sum is $-X$, an event that occurs almost-surely infinitely often when an outcome π is such that $\mathsf{Inf}(\pi) = U$ for some EC $U \in \mathcal{L} = \mathcal{E} \setminus \mathcal{W}$, $\mathcal{P}_1$ will *believe* its running sum hits zero, *whether it really does or not*. Consequently, he has to leave U_1 to ensure the worst-case requirement at some point.

Optimal expected values can be reached in winning end-components. Consider a setting satisfying Assumption 1: a game G reduced to a winning EC such that $E_\Delta = E$. Let the worst-case threshold be $\mu = 0$, as usual. In Sect. 4.6, we have seen that, for all $\varepsilon > 0$, it is possible to combine a worst-case strategy λ_1^{wc} with an optimal expectation strategy λ_1^e into a finite-memory strategy λ_1^{cmb} that ensures satisfaction of the BWC problem for thresholds $(0, \nu^* - \varepsilon)$, where ν^* is the maximal expected value in $P = G[\lambda_2^{\mathsf{stoch}}]$. Observe that in general, it is not possible to construct a finite-memory strategy λ_1^{cmb} that ensures the worst-case while inducing an expected value exactly equal to ν^* against the stochastic model. See for example the game $G \restriction U_3$ in Fig. 2: clearly, $\mathcal{P}_1$ has to use (s_{10}, s_9) infinitely often to ensure the worst-case, and when using finite memory, the contribution (in terms of proportion of cycles played) of the corresponding cycle in the overall expectation can be lower bounded by a strictly positive probability, hence inducing an expected value strictly lower than $\nu^* = 2$.

Nonetheless, it is possible to build an infinite-memory strategy, denoted λ_1^{inf}, that exactly achieves this expectation while verifying the worst-case threshold. It is in essence similar to the finite-memory combined strategy (Def. 3). Consider the following argument. Observe that in the analysis of the combined strategy (Sect. 4.6), we show that when parameters K and $L(K)$ tend to infinity, the expectation induced by λ_1^{cmb} tends to ν^*. Moreover, the worst-case is always ensured by choice of $L(K)$. Hence, we possess all the elements needed to construct λ_1^{inf}: it suffices to implement a strategy that plays as λ_1^{cmb}, but sequentially increasing the values of K and $L(K)$ up to infinity. Formally, let $(K_i)_{i \in \mathbb{N}}$ be a strictly increasing sequence of naturals, and for all $i \geq 0$, let $L(K_i)$ be the natural given by Def. 4. The strategy λ_1^{inf} is defined as follows:

(*init*) Initialize i to 0.
(a) Play λ_1^e for K_i steps and memorize $\mathsf{Sum} \in \mathbb{Z}$, the sum of weights encountered during these K_i steps.

(b) If $\mathsf{Sum} > 0$, then go to *(a)*.
Else, play λ_1^{wc} during $L(K_i)$ steps, then increment i and go to *(a)*.

By doing so, it is possible to show that λ_1^{inf} ensures an expected mean-payoff exactly equal to ν^*, as well as the worst-case requirement. For the worst-case, it suffices to apply the reasoning developed in Lemma 10. To show that λ_1^{inf} achieves the expected value ν^*, the intuitive argument is that the probability that a period of type *(a)* is followed by a period of type *(b)* tends to zero as K_i grows, since $\nu^* > 0$. Therefore, the probability that λ_1^{wc} is played infinitely many times is zero.

To illustrate this point, let us consider the example of Fig. 2. By playing λ_1^{inf} as defined above, $\mathcal{P}_1$ can ensure the worst-case requirement and induce the optimal expected mean-payoff 2, because the proportion of time spent following strategy λ_1^e will tend to one as the parameter K_i tends to infinity.

5 Truncated Sum Value Function - Shortest Path Problem

Let us consider a game $G = (\mathcal{G}, S_1, S_2)$ with an underlying graph $\mathcal{G} = (S, E, w)$ such that the weight function $w\colon E \to \mathbb{N}_0$ assigns *strictly positive* integer weights to all edges, and a target set of states $T \subseteq S$ that $\mathcal{P}_1$ wants to reach with a path of bounded value. That is, $\mathcal{P}_1$ aims to ensure some threshold on the *truncated sum* value function TS_T. In other words, we study the BWC synthesis problem for the *shortest path problem* [2,11]. More precisely, given an initial state $s_{\mathsf{init}} \in S$, the goal of $\mathcal{P}_1$ is to ensure to reach T with a path of (truncated) sum strictly lower than the threshold $\mu \in \mathbb{N}$ (we assume a natural threshold w.l.o.g. as all weights take positive integer values and so does the truncated sum function for any play reaching T) against all possible behaviors of $\mathcal{P}_2$ while guaranteeing, at the same time, an expected cost to target strictly lower than the threshold $\nu \in \mathbb{Q}$ against the finite-memory stochastic model of the adversary specified by the stochastic Moore machine $\mathcal{M}(\lambda_2^{\mathsf{stoch}})$. Hence, notice that regarding Def. 1, the inequalities are reversed. Hence we assume that $\nu < \mu$. Equivalently, the problem could be stated with value function $-\mathsf{TS}_T$ without changing the definition.

We provide here the following results. First, we show that satisfaction of the BWC problem for the truncated sum value function can be decided in pseudo-polynomial time (Sect. 5.1). Second, we show that pseudo-polynomial memory may be necessary to satisfy the BWC problem and that it always suffices (Sect. 5.2). Third, we show that the decision problem cannot be solved in polynomial time unless P = NP by providing an NP-hardness result (Sect. 5.3).

5.1 A pseudo-polynomial time algorithm

To solve the decision problem, we proceed as follows. First, we show how to construct, from the original game G and the worst-case threshold μ, a new game G_μ such that there is a one-to-one correspondence between the strategies of $\mathcal{P}_1$ in G_μ and the strategies of $\mathcal{P}_1$ in the original game G that are winning for the worst-case requirement (eq. (1)). To construct this game, we unfold the original graph $\mathcal{G}$, tracking the current value of the truncated sum *up to the worst-case threshold* μ, and integrating this value in the states of an expanded graph $\mathcal{G}'$. In the corresponding game G', we then compute the set of states R from which $\mathcal{P}_1$ can reach the target set with cost lower than the worst-case threshold and we define the subgame $G_\mu = G' \restriction R$ such that any path in the graph of G_μ satisfies the worst-case requirement. Second, from this new game G_μ and the stochastic Moore machine $\mathcal{M}(\lambda_2^{\mathsf{stoch}})$ representing the stochastic model of the adversary, we construct an MDP in which we search for a $\mathcal{P}_1$ strategy that ensures reachability of T with an expected cost strictly lower than ν (eq. (2)). If such a strategy exists, it is guaranteed that it will also satisfy the worst-case requirement against any strategy of $\mathcal{P}_2$ thanks to the bijection evoked earlier.

Theorem 5. *The beyond worst-case problem for the shortest path can be solved in pseudo-polynomial time: polynomial in the size of the underlying game graph, the Moore machine for the stochastic model of the adversary and the encoding of the expected value threshold, and polynomial in the value of the worst-case threshold.*

Proof. Let $G = (\mathcal{G}, S_1, S_2)$ be the two-player game, $\mathcal{G} = (S, E, w)$ its underlying graph, $w\colon E \to \mathbb{N}_0$ its weight function, $s_{\mathsf{init}} \in S$ the initial state, $T \subseteq S$ the target set, $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^F$ the stochastic model of $\mathcal{P}_2$, with $\mathcal{M}(\lambda_2^{\mathsf{stoch}}) = (\mathsf{Mem}, \mathsf{m}_0, \alpha_{\mathsf{u}}, \alpha_{\mathsf{n}})$ its Moore machine, $\mu \in \mathbb{N}$ the worst-case threshold, and $\nu \in \mathbb{Q}$ the expected value threshold.

Based on G and μ, we define the game $G' = (\mathcal{G}', S'_1, S'_2)$. Its underlying graph $\mathcal{G}' = (S', E', w')$ is built by unfolding the original graph $\mathcal{G}$, tracking the current value of the truncated sum *up to the worst-case threshold* μ, and integrating this value in the states of $\mathcal{G}'$. Formally, we have that

- $S'_1 = S_1 \times (\{0,1,\ldots,\mu-1\} \cup \{\top\})$, $S'_2 = S_2 \times (\{0,1,\ldots,\mu-1\} \cup \{\top\})$, and $S' = S'_1 \cup S'_2$;
- $E' = \big\{\big((s_1,u_1),(s_2,u_2)\big) \in S' \times S' \mid (s_1,s_2) \in E \wedge u_2 = u_1 + w((s_1,s_2))\big\}$, with the convention that, for all $c \in \mathbb{N}$, $\top + c = \top$, and, for all $u \in \mathbb{N}$, $u + c = \top$ if $u + c \geq \mu$;
- $\forall e = \big((s_1,u_1),(s_2,u_2)\big) \in E'$, $w'(e) = w((s_1,s_2))$.

The symbol $\top$ represents costs exceeding the worst-case threshold μ. The initial state in G' is $s'_{\mathsf{init}} = (s_{\mathsf{init}}, 0)$, and the target set is $T' = T \times \{0,1,\ldots,\mu-1\}$, i.e., in G' the target set is restricted to copies of states of the original target set that are reached with a sum less than μ. Notice that for any state $s'_1 = (s_1, \top) \in S'$, all its successors in $\mathcal{G}'$ are of the form $s'_2 = (s_2, \top)$.

Now, we compute in G' the set of states $R \subseteq S'$ from which $\mathcal{P}_1$ has a strategy to force reaching T' using a classical attractor computation, i.e., $R = \mathsf{Attr}^{\mathcal{P}_1}_{G'}(T')$ (cf. Sect. 2). Clearly, all states outside this attractor set are losing for the worst-case requirement. Indeed, from any state outside of R, either $\mathcal{P}_1$ cannot force reaching a state $s' = (s,u)$ with $s \in T$, or he can only do it for $u = \top$. In particular, if $s'_{\mathsf{init}} = (s_{\mathsf{init}}, 0) \notin R$ then we know that $\mathcal{P}_1$ cannot enforce the worst-case threshold in the original game G, and we can stop here in this case: no strategy exists for the BWC problem.

Assume that $s'_{\mathsf{init}} = (s_{\mathsf{init}}, 0) \in R$. Let us write $G_\mu = G' \restriction R$. Note that there will be deadlocks in G_μ (i.e., states with no successors): this is guaranteed since the sum of weights is strictly growing (recall $w\colon E \to \mathbb{N}_0$) and states of the form $(s, \top)$ do not belong to R by definition. However, the only deadlocks will be on states that are in $T' \subseteq R$ (by definition of R as the attractor of T'). Hence, we easily get rid of them by adding self-loops of weight zero (this does not change the truncated sum of paths up to T' by definition of $\mathsf{TS}_{T'}$). It is easy to see that all strategies $\lambda_1 \in \Lambda_1(G_\mu)$ are winning for the worst-case requirement, and that there is a bijection between the winning strategies for the worst-case requirement in the original game G and the strategies in G_μ.

We are now equipped to handle the expectation objective. We proceed as follows. First, we take the product of G_μ and $\mathcal{M}(\lambda_2^{\mathsf{stoch}}) = (\mathsf{Mem}, \mathsf{m}_0, \alpha_{\mathsf{u}}, \alpha_{\mathsf{n}})$, following the construction of Lemma 4 (the proof holds for the truncated sum value function as well). On the product game, we again preserve correspondence with the worst-case winning strategies in the original game G. Applying the memoryless stochastic model (resulting of Lemma 4) on the product game, we obtain an MDP P. It is then clear that $\mathcal{P}_1$ has a strategy to enforce an expected value strictly less than the threshold ν in P *if and only if* $\mathcal{P}_1$ has a strategy that *both* enforces the worst-case threshold against any strategy $\lambda_2 \in \Lambda_2(G)$, and the expectation threshold against $\lambda_2^{\mathsf{stoch}}$ in G. To decide if such a strategy exists, we compute the minimal achievable expected value on P and we compare it against the threshold ν. Thanks to the reduction from truncated sum to total-payoff proposed in the preliminaries (Sect. 2), we know that this optimal value can be achieved by a memoryless strategy and its computation can be executed in polynomial time in the size of the encoding of P via linear programming [14]. Hence, it requires time polynomial in the size of the encoding of G and $\mathcal{M}(\lambda_2^{\mathsf{stoch}})$, and polynomial in the value μ (since $|S'| = |S| \cdot (\mu+1)$). □

5.2 Memory requirements

In Thm. 6, we characterize the memory needed by strategies satisfying the BWC problem. The construction developed in Thm. 5 yields an upped bound for the memory that is polynomial in the size of the game and the stochastic model, and in the value of the worst-case threshold. Indeed, the synthesized strategy is memoryless in the MDP P that is obtained by taking the product of the expanded game G_μ, such that $|G_\mu| \leq |G| \cdot (\mu+1)$, with the Moore machine $\mathcal{M}(\lambda_2^{\mathsf{stoch}})$. Hence the memory needed is bounded by a polynomial in the sizes $|G|$ and $|\mathcal{M}(\lambda_2^{\mathsf{stoch}})|$, and in the value μ.

We also exhibit a family of games (Fig. 7) for which winning the BWC problem requires memory linear in μ, hence proving that the pseudo-polynomial bound is tight. The intuition is as follows. Assume a worst-case threshold $\mu \in \{13 + k \cdot 4 \mid k \in \mathbb{N}\}$ (the

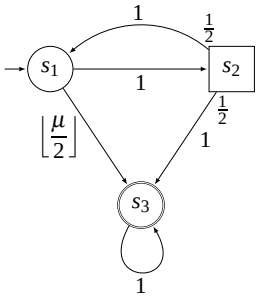

Fig. 7: Family of games $(G(\mu))_{\mu \in \{13+k\cdot 4 \mid k \in \mathbb{N}\}}$ requiring memory linear in μ for the BWC problem.

set is defined in order to ease computations). From state s_1, $\mathcal{P}_1$ can ensure reaching the target set $T = \{s_3\}$ at a guaranteed cost of $\lfloor \frac{\mu}{2} \rfloor$. Nevertheless, in order to *minimize* the expected cost of reaching T, $\mathcal{P}_1$ should try to reach it via state s_2, as the cost will be diminished. Hence, $\mathcal{P}_1$ should play edge (s_1, s_2) repeatedly, up to the point where playing (s_1, s_3) becomes mandatory to preserve the worst-case requirement (i.e., when the running sum of weights becomes equal to $\lfloor \frac{\mu}{2} \rfloor$ as the total cost for the worst outcome will be $2 \cdot \lfloor \frac{\mu}{2} \rfloor < \mu$). To implement this strategy (Fig. 8), $\mathcal{P}_1$ has to play (s_1, s_2) exactly $\lfloor \frac{\mu}{4} \rfloor$ times and then switch to (s_1, s_3). Clearly, this requires memory linear in μ. The expected value threshold ν can be chosen sufficiently low so that $\mathcal{P}_1$ is compelled to use this optimal strategy to satisfy the BWC problem.

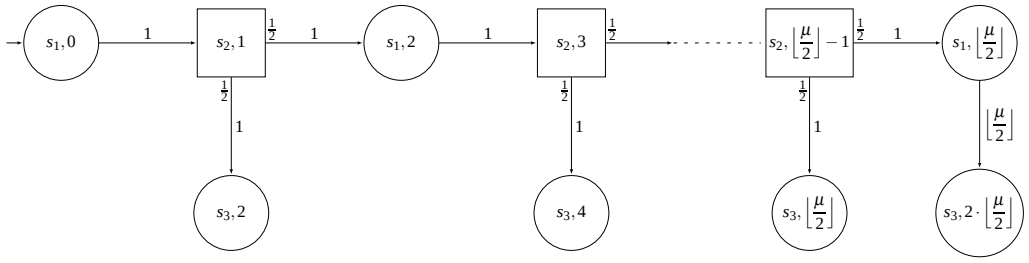

Fig. 8: Partial representation of the Markov chain induced by the BWC strategy that minimizes the expected cost to target in $G(\mu)$, $\mu \in \{13 + k \cdot 4 \mid k \in \mathbb{N}\}$.

Theorem 6. *Memory of pseudo-polynomial size may be necessary and is always sufficient to satisfy the BWC problem for the shortest path: polynomial in the size of the game and the stochastic model, and polynomial in the worst-case threshold value.*

Proof. The upper bound on the size of the memory can be obtained directly from the construction exposed in the proof of Thm. 5. Indeed, we have shown that if the BWC problem can be satisfied, the memoryless strategy that minimizes the expectation in the MDP P does satisfy it. Translated back to the original game, this strategy has a memory which is polynomial in $|G|$, $|\mathcal{M}(\lambda_2^{\mathsf{stoch}})|$, and the value of μ. Intuitively, the strategy needs to memorize the current value of the sum of weights, up to the value of the worst-case threshold (at which point it does not matter to bookkeep it anymore as $\mathcal{P}_1$ has already failed to enforce the worst-case requirement). Hence, such a strategy requires memory polynomial in the input game and the stochastic model, and in the threshold.

To prove that pseudo-polynomial memory may be necessary, we introduce a family of games $(G(\mu))_{\mu \in \{13+k\cdot 4 \mid k \in \mathbb{N}\}}$, indexed by the value of the worst-case threshold. This value is taken in a specific set $\{13 + k \cdot 4 \mid k \in \mathbb{N}\}$ mostly to ease the following calculations. The family is presented in Fig. 7: it consists of three states $S = \{s_1, s_2, s_3\}$. The weight function only assigns strictly positive weights as assumed in the setting of the shortest path problem. All weights are equal to 1 except for edge (s_1, s_3) which has a weight $\lfloor \frac{\mu}{2} \rfloor$. Notice that μ is chosen odd and such that $\lfloor \frac{\mu}{2} \rfloor$ is even.

We will consider the values of the expectation threshold ν that can be ensured by a BWC strategy in such a game, under the chosen worst-case threshold μ and against a stochastic model assigning uniform distributions, and show that to minimize this value, $\mathcal{P}_1$ needs to use linear memory in μ, hence proving the claim.

First, remark that if the running sum of weights (which is an integer value) gets strictly larger than $\lfloor \frac{\mu}{2} \rfloor$, then $\mathcal{P}_1$ has lost the worst-case requirement (eq. (1)) as playing (s_1, s_2) does not guarantee reaching T, and playing (s_1, s_3) induces a total cost at least equal to μ. Hence, when in s_1 with a running sum equal to $\lfloor \frac{\mu}{2} \rfloor$, $\mathcal{P}_1$ has no valid choice but to take the edge (s_1, s_3). Since randomization clearly does not help (as it will produce consistent outcomes that are losing if the edge (s_1, s_2) is repeatedly assigned a non-zero probability), defining the optimal strategy of $\mathcal{P}_1$ boils down to deciding for how long he should take the edge (s_1, s_2) before switching (if at all).

We claim that it should maximize the number of passes in s_2 (Fig. 8). Let n denotes the number of times $\mathcal{P}_1$ chooses (s_1, s_2) before switching. Clearly, to guarantee satisfaction of the worst-case requirement, we need $2 \cdot n + \lfloor \frac{\mu}{2} \rfloor < \mu$. Since the threshold is odd, we have $2 \cdot n + \lfloor \frac{\mu}{2} \rfloor < 2 \cdot n + \frac{\mu}{2}$. Hence, it suffices to have $2 \cdot n + \frac{\mu}{2} \leq \mu$, or equivalently, $n \leq \frac{\mu}{4}$. Note that this bound is linear in the value of μ. What remains to prove is that increasing the number of passes

results in a decrease of the expected value. Let $e(n)$ denotes the expected value induced by the strategy that plays edge (s_1, s_2) for n times before switching. Careful computation reveals that $e(n)$ can be expressed as follows:

$$e(n) = \sum_{i=0}^{n-1} \frac{1}{2^{i-1}} + \frac{1}{2^n} \cdot \left\lfloor \frac{\mu}{2} \right\rfloor . \tag{17}$$

Our thesis is that for all $n \geq 0$, $e(n) < e(n-1)$. By eq. (17), that is

$$\begin{gathered}
\sum_{i=0}^{n-1} \frac{1}{2^{i-1}} + \frac{1}{2^n} \cdot \left\lfloor \frac{\mu}{2} \right\rfloor < \sum_{i=0}^{n-2} \frac{1}{2^{i-1}} + \frac{1}{2^{n-1}} \cdot \left\lfloor \frac{\mu}{2} \right\rfloor , \\
\frac{1}{2^{n-2}} - \frac{1}{2^n} \cdot \left\lfloor \frac{\mu}{2} \right\rfloor < 0, \\
\left\lfloor \frac{\mu}{2} \right\rfloor > \frac{2^n}{2^{n-2}} = 4, \\
\mu > 9,
\end{gathered}$$

and the last inequality is granted thanks to the hypothesis that $\mu \in \{13 + k \cdot 4 \mid k \in \mathbb{N}\}$. This shows that increasing n decreases the expectation, as wanted.

In conclusion, the optimal BWC strategy for the expected value criterion consists in playing (s_1, s_2) for exactly $n = \left\lfloor \frac{\mu}{4} \right\rfloor$ times, then swithing to (s_1, s_3) to ensure the worst-case (the corresponding MC is represented in Fig. 8). Following our computations, it is possible to impose that playing this strategy is necessary to satisfy the BWC problem by taking the expected value threshold such that $e(n) < \nu \leq e(n-1)$. This proves that memory linear in μ is needed for the given family of games. □

Remark 10. In contrast to the case of the mean-payoff value function (Sect. 4.11), infinite memory gives no additional power in the shortest path context. Indeed, the proof of Thm. 5 gives a complete representation of worst-case winning strategies through the game G_μ and it is further proved that finite memory suffices to define an optimal strategy with regard to the expected value criterion among these worst-case winning strategies.

5.3 NP-hardness of the decision problem

We conclude our study of the BWC problem in the shortest path setting (i.e., for the truncated sum value function) by showing that it is very unlikely that a truly-polynomial (i.e., also polynomial in the size of the encoding of the worst-case threshold) time algorithm exists, as we establish in Thm. 7 that the decision problem is NP-hard. Actually, it is likely that the problem is not in NP at all, since we prove a reduction from the K^{th} *largest subset problem* which is known to be NP-hard and commonly thought to be outside NP as natural certificates for the problem are larger than polynomial [16].

The K^{th} largest subset problem is expressed as follows. Given a finite set A, a size function $h\colon A \to \mathbb{N}_0$ assigning strictly positive integer values to elements of A, and two naturals $K, L \in \mathbb{N}$, decide if there exist K distinct subsets $C_i \subseteq A$, $1 \leq i \leq K$, such that $h(C_i) = \sum_{a \in C_i} h(a) \leq L$ for all K subsets. The NP-hardness of this problem was proved in [22] via a Turing reduction from the partition problem.

The key steps of the reduction are as follows. We build a game composed of two gadgets. The *random subset selection gadget* (Fig. 9) stochastically generates paths that represent subsets of A. It has the important property that all subsets are equiprobable. The *choice gadget* follows (Fig. 10). In it, $\mathcal{P}_1$ decides either to go to s_e, which leads to lower expected values (and lower is better in our setting) but may be dangerous for the worst-case requirement, or to go to s_{wc}, which is always safe with regard to the worst-case threshold but induces an higher expected cost. The trick is to prove that we can define values of the thresholds and the weights used in the gadgets such that an optimal[15] strategy for $\mathcal{P}_1$ consists in choosing state s_e only when the randomly generated subset $C \subseteq A$ satisfies $h(C) \leq L$, as asked by the K^{th} largest subset problem; and such that this strategy satisfies the BWC problem if and only if there exist K distinct subsets that verify this bound, i.e., if and only if the answer to the K^{th} largest subset problem is YES.

[15] Minimizing the expectation while guaranteeing a given worst-case threshold.

Theorem 7. *The beyond worst-case problem for the shortest path is NP-hard.*

Proof. We establish a reduction from the K^{th} largest subset problem: given a finite set $A = \{a_1, \ldots, a_n\}$ (hence $n = |A|$), a size function $h\colon A \to \mathbb{N}_0$, and two naturals $K, L \in \mathbb{N}$, decide if there exist K distinct subsets $C_i \subseteq A$, $1 \leq i \leq K$, such that $h(C_i) = \sum_{a \in C_i} h(a) \leq L$ for all K subsets. This problem is known to be NP-hard [22,16]. Note that the restriction to $\mathbb{N}_0$ for the codomain of h in place of $\mathbb{N}$ is w.l.o.g. as the problem is satisfied for A, K, L and $h\colon A \to \mathbb{N}$ if and only if it is satisfied for $A' = A \setminus \{a \in A \mid h(a) = 0\}$, $K' = \left\lfloor \frac{K}{2^{|A|-|A'|}} \right\rfloor$, $L' = L$ and $h'\colon A' \to \mathbb{N}_0$ such that for all $a \in A'$, $h'(a) = h(a)$. Remark that, obviously, we should have $K \leq 2^n$, otherwise the problem is trivial since we cannot find a sufficient number of *distinct* subsets.

Before giving the details of our reduction, we define, given A and h, the function $h_n\colon A \to \mathbb{N}_0$ such that for each $a \in A$, $h_n(a) = (n+1) \cdot h(a)$. Clearly, it satisfies the following property:

$$\forall C \subseteq A, h(C) \leq L \Leftrightarrow h_n(C) \leq (n+1) \cdot L. \tag{18}$$

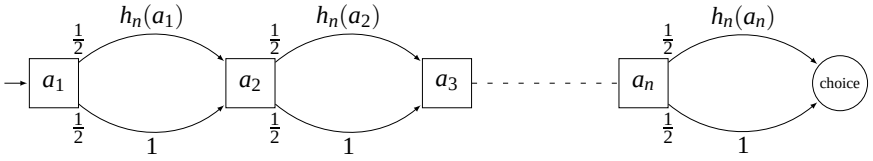

Fig. 9: Random subset selection gadget: an element is selected in the subset if the upper edge is taken when leaving the corresponding state.

We now present two gadgets useful to construct a game and an associated BWC shortest path problem such that the answer to the K^{th} largest subset problem is YES if and only if the answer to the BWC problem is also YES.

First, the fragment of the game graph depicted in Fig. 9 is called the *random subset selection gadget*. All its states belong to $\mathcal{P}_2$, except for the last one, and model the selection (or not) of an element of A in a subset. Basically, there is a bijection between paths[16] in this gadget and subsets of A: an element $a_i \in A$ is selected by the gadget if the outgoing upper edge is taken when leaving state a_i, and not selected when the outgoing lower edge is taken. The stochastic model followed by $\mathcal{P}_2$ in the BWC shortest path problem we construct is the uniform distribution: the upper and lower edges are equiprobable in all states. This gadget verifies the following important properties.

1. All subsets are equiprobable: they have probability $\frac{1}{2^n}$ to be selected.
2. If the gadget selects a subset $C \subseteq A$ through the corresponding path p_C, the total sum of weights along p_C, denoted by $t(p_C)$, is equal to $h_n(C) + n - |C|$.

By eq. (18), we have that

$$\forall C \subseteq A, h(C) = L \Leftrightarrow (n+1) \cdot L \leq t(p_C) < (n+1) \cdot (L+1). \tag{19}$$

Indeed, consider the following. Remark that $0 \leq n - |C| \leq n$, for any subset $C \subseteq A$. Hence the left-to-right implication is trivial. For the converse, we directly deduce the following equivalent expression:

$$L - \frac{n-|C|}{n+1} \leq h(C) < L + 1 - \frac{n-|C|}{n+1}.$$

The left inequality implies that $L - 1 < h(C)$, and since $h(C) \in \mathbb{N}$, that $L \leq h(C)$. The right inequality implies that $h(C) < L + 1$, and using the same argument, that $h(C) \leq L$. We conclude that eq. (19) is true. Consequently, we define the value $T = (n+1) \cdot (L+1) - 1$, which is an upper bound on the value $t(p_C) \leq T$ of a path corresponding to a subset $C \subseteq A$ such that $h(C) \leq L$.

[16] To be able to formally distinguish between such paths, which we usually define as sequence of *states*, we should introduce dummy states to split edges. We omit this technical trick for the sake of simplicity.

Now consider the second gadget, called the *choice gadget* and depicted in Fig. 10. This gadget comes after the random subset selection gadget. Its aim is to discriminate subsets generated by the preceding gadget based on whether or not they satisfy the upper bound $h(C) \leq L$. Observe the shared *choice* state. There, $\mathcal{P}_1$ has the choice to go up to state s_e or down to state s_{wc}. Both belong to $\mathcal{P}_2$. Again, probabilities for the stochastic model of the adversary are depicted in Fig. 10. So, in s_e, an arbitrary strategy of $\mathcal{P}_2$ can decide to impose cost x_1 or cost x_2 before reaching the target set of the game (notice we have set the weight of the self-loop to zero on the target set, as discussed previously). Nonetheless, the stochastic model $\lambda_2^{\mathsf{stoch}}$ of $\mathcal{P}_2$ assigns probability zero to the edge of weight x_1: the expectation of any strategy of $\mathcal{P}_1$ against this stochastic model will be independent of the value x_1. In s_{wc}, the cost added is always equal to x_3. Intuitively, we will choose values so that to minimize its expected cost-to-target, $\mathcal{P}_1$ should choose s_e, but also so that the worst-case requirement implies that it is only safe to choose this state if the previous path defined a subset that satisfies the bound $h(C) \leq L$ given by the K^{th} largest subset problem.

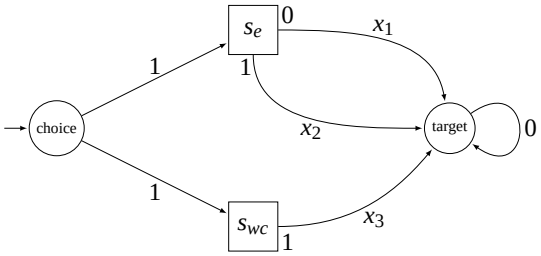

Fig. 10: Choice gadget: choosing s_e is best for the expected value, but it is safe with regard to the worst-case if and only if the random subset selection produced a subset C such that $h(C) \leq L$.

We now have defined the game graph (Fig. 9 and 10) and the stochastic model for the BWC problem. To complete the description of the reduction, we need to precise the values of the thresholds μ and ν, and the weights x_1, x_2 and x_3. Assume that we choose the worst-case threshold and the weights such that:

(a) $T + 1 + x_1 + 1 \geq \mu$, i.e., going to s_e with a path p_C (obtained in the random subset selection gadget) of cost $t(p_C) > T$ (i.e., with a selected subset $C \subseteq A$ such that $h(C) > L$ by eq. (19)) is losing for the worst-case threshold if $\mathcal{P}_2$ takes the edge of weight x_1;
(b) $T + x_1 + 1 < \mu$ and $T + x_2 + 1 < \mu$, i.e., going to s_e with a path p_C of cost $t(p_C) \leq T$ (i.e., $h(C) \leq L$) is safe for the worst-case requirement whatever the choice of $\mathcal{P}_2$;
(c) for all $C \subseteq A$, we have that $t(p_C) + x_3 + 1 < \mu$, i.e., going to s_{wc} is always safe for the worst-case requirement.

Then clearly, $\mathcal{P}_1$ can always choose to go to s_{wc} and ensure the worst-case threshold, but he can go up only if the chosen subset C satisfies $h(C) \leq L$, which is equivalent to say that $t(p_C) \leq T$. We add the following constraints to the choices of the expectation threshold and the weights:

(d) in order to minimize the expected truncated sum in the MDP defined by the stochatic model, the optimal choice for $\mathcal{P}_1$ is to always take s_e when possible (i.e., when $h(C) \leq L$, or equivalently $t(p_C) \leq T$ because of the constraint *(a)* defined above);
(e) the expected value v^* of this optimal choice satisfies the expectation requirement (i.e., $v^* < \nu$) if and only if the number of distinct subsets $C_i \subseteq A$ verifying $h(C_i) \leq L$ is larger than or equal to K.

We will now define values such that properties *(a)* through *(e)* are ensured. First, let $Q = \max\{t(p_C) \mid C \subseteq A\} = t(p_A)$ be the maximal cost of a path in the random subset selection gadget (the equality with $t(p_A)$ is thanks to the size function h assigning strictly positive values). We claim the needed properties are verified for the following values:

$$\mu = 2^{n+1} \cdot n \cdot (Q+2), \qquad \nu = \frac{K \cdot (T+2) + (2^n - K) \cdot \mu}{2^n},$$

$$x_1 = \mu - T - 2, \qquad x_2 = 1, \qquad x_3 = \mu - Q - 2.$$

Using these, we review each property one-by-one. For *(a)*, we obtain by simple substitutions

$$(a) \quad \Leftrightarrow \quad T+1+\mu-T-2+1 \geq \mu \quad \Leftrightarrow \quad 0 \geq 0,$$

which is obviously true. Similarly, for *(b)*, we have that

$$(b) \quad \Leftrightarrow \quad (T+\mu-T-2+1<\mu) \wedge (T+1+1<\mu) \quad \Leftrightarrow \quad (-1<0) \wedge (T+2<2^{n+1}\cdot n\cdot(Q+2)). \tag{20}$$

The first term of the conjunction is trivially true so we focus on the second one. Without loss of generality, we can assume that $L < h(A)$ as otherwise the K^{th} largest subset problem reduces to decide if $K \leq 2^n$. Thus, we deduce the inequality $T < (n+1)\cdot(h(A)+1)-1$. Also note that, by definition, we have that $Q = t(p_A) = h_n(A) = (n+1)\cdot h(A)$. Using these inequalities in eq. (20), we derive that proving the following central inequality suffices to obtain *(b)*:

$$T+2 < (n+1)\cdot(h(A)+1)+1 \leq 2^{n+1}\cdot n\cdot((n+1)\cdot h(A)+2) = 2^{n+1}\cdot n\cdot(Q+2).$$

This boils down to

$$(2^{n+1}\cdot n-1)\cdot(n+1)\cdot h(A)+(2\cdot 2^{n+1}-1)\cdot n-2 \geq 0,$$

which is true for $n \geq 1$ (which we can assume otherwise $A = \emptyset$ and the problem is trivial). Hence, property *(b)* is verified by our choice of values. Now, consider property *(c)*. We have

$$(c) \quad \Leftrightarrow \quad t(p_C)+\mu-Q-2+1<\mu \quad \Leftrightarrow \quad t(p_C)<Q+1,$$

which is true by definition of Q as the maximum over the values of paths. Regarding property *(d)*, we have to show that choosing s_e gives an expectation strictly lower (recall we want to minimize it) than choosing s_{wc}. Observe that due to the particular structure of the game graph, the strategy of $\mathcal{P}_1$ is restricted to this one-shot choice of edge. Remark that in this expected value context, the actual value obtained in the random subset selection gadget does not matter to decide whether to go to s_e or to s_{wc}: hence it suffices to look at the expectation from the *choice* state up to the *target* state. For s_e, it is trivially equal to $1+1=2$ as the stochastic model $\lambda_2^{\mathsf{stoch}}$ of $\mathcal{P}_2$ always chooses the edge of weight x_2. For s_{wc}, this expectation is equal to

$$1+x_3 = 1+\mu-Q-2 = 2^{n+1}\cdot n\cdot(Q+2)-Q-1 \geq (2^{n+1}\cdot n-1)\cdot(Q+2) \geq (Q+2) > 2,$$

and we obtain the claim *(d)*. Note that an actual strategy that satisfies the BWC problem will only be able to choose s_e if the selected path satisfies the bound $t(p_C) \leq T$, as discussed in properties *(a)* and *(b)*.

Finally, it remains to show the most involved property *(e)*: proving it will conclude our reduction as we will obtain that the answer to the K^{th} largest subset problem is YES if and only if the answer to the BWC problem we have defined is YES. Note that combining the already proved properties *(a)* through *(d)*, we know that the strategy $\lambda_1 \in \Lambda_1^{PF}$ that chooses state s_e when $t(p_C) \leq T$ and state s_{wc} otherwise, yields the minimal expectation value v^* *under the worst-case constraint* of threshold μ. Hence, it suffices to study this strategy to answer the BWC problem. Our claim is thus that

$$v^* = \mathbb{E}_{a_1}^{G[\lambda_1,\lambda_2^{\mathsf{stoch}}]} < v \quad \Leftrightarrow \quad \Big|\{C \subseteq A \mid h(C) \leq L\}\Big| \geq K, \tag{21}$$

with G and $\lambda_2^{\mathsf{stoch}}$ the game and stochastic model we defined.

For the left-to-right implication, we reason by contradiction and show that if there is only $K-1$ (or less) distinct subsets whose sum is less than or equal to L, then strategy λ_1 has an expected cost larger than or equal to v. To show that, we use the fact that all paths (i.e., subsets) have equal probability in the random subset selection gadget, and establish that a lower bound on the sum of all the paths under this strategy reaches or exceeds $2^n \cdot v$. Recall that $\mathcal{P}_2$ follows its stochastic model $\lambda_2^{\mathsf{stoch}}$ for this matter. First, let $\mathsf{LB}_e = 0$ which is trivially a lower bound for the cost of all the paths that goes through s_e. Second, let $\mathsf{LB}_{wc} = (2^n-(K-1))\cdot(T+1+x_3+1)$: it is clearly a lower bound for the sum of the values of paths that go through state s_{wc} when $\mathcal{P}_1$ follows strategy λ_1. We have that $2^n \cdot v^* \geq \mathsf{LB}_e + \mathsf{LB}_{wc}$. Let us now establish that $\mathsf{LB}_{wc} \geq 2^n \cdot v$ and we will be done. We proceed as follows.

$$\begin{aligned}
\mathsf{LB}_{wc} - 2^n\cdot v &= (2^n-K+1)\cdot(T+1+\mu-Q-2+1)-K\cdot(T+2)-(2^n-K)\cdot\mu \\
&= \mu+(2^n-K+1)\cdot(T-Q)-K\cdot(T+2) \\
&= 2^{n+1}\cdot n\cdot(Q+2)+(2^n-K+1)\cdot(T-Q)-K\cdot(T+2)
\end{aligned}$$

Recall that $T, Q \geq 0$, $n \geq 1$ and $0 \leq K \leq 2^n$ (otherwise the answer is trivially NO). Furthermore, T is the upper bound on the values of paths p_C representing good subsets, i.e., subsets $C \subseteq A$ such that $h(C) \leq L$. This value is used by the strategy λ_1 implemented by $\mathcal{P}_1$ to decide whether going to s_e is safe with regard to the worst-case requirement or not. As such, we can assume that $T \leq Q$, otherwise all paths are safe and the answer to the problem is trivially YES (since all subsets respect the bound and $K \leq 2^n$). Using $K \leq 2^n$, $T \geq 0$ and $T \leq Q$, we can write

$$\mathsf{LB}_{wc} - 2^n \cdot \nu \geq (2^{n+1} \cdot n - 2^n + K - 1 - K) \cdot Q + (2^{n+1} \cdot n \cdot 2 - 2 \cdot K). \tag{22}$$

To prove that this last expression is non-negative, we analyze its terms. We know that $Q \geq 0$. For its coefficient, we have

$$2^{n+1} \cdot n - 2^n - 1 \geq 2^n - 1 \geq 0$$

because $n \geq 1$. For the last term, we use $K \leq 2^n$ and obtain that

$$2^{n+2} \cdot n - 2 \cdot K \geq 2^{n+2} - 2^{n+1} = 2^{n+1} \geq 0.$$

Hence all terms of eq. (22) are non-negative and $\mathsf{LB}_{wc} \geq 2^n \cdot \nu$, proving that the left-to-right implication of eq. (21) is verified.

It remains to prove the right-to-left implication. Assume there are exactly K distinct subsets of sum less than or equal to L (if there are more, then the bounds below are easier to obtain). We claim that strategy λ_1 ensures an expected truncated sum ν^* strictly lower than ν. To show this, we establish that the total sum of the outcomes under this strategy of $\mathcal{P}_1$ and the stochastic model of $\mathcal{P}_2$ is strictly bounded from above by $2^n \cdot \nu$, and the claim follows thanks to all paths being equiprobable in the random subset selection gadget. First, consider the paths that go through s_e (i.e., all the paths corresponding to subsets C such that $h(C) \leq L$). By definition of λ_1 and our hypothesis, there are exactly K such paths. We define $\mathsf{UB}_e = K \cdot (T+2)$, a clear upper bound for the sum of the values of these paths, by definition of T and $\lambda_2^{\mathsf{stoch}}$. Second, there are $(2^n - K)$ paths that go through s_{wc}. Let $\mathsf{UB}_{wc} = (2^n - K) \cdot (Q + 1 + x_3) = (2^n - K) \cdot (\mu - 1)$ be a bound for the sum of the values of all these paths. Clearly,

$$2^n \cdot \nu^* \leq \mathsf{UB}_e + \mathsf{UB}_{wc} = K \cdot (T+2) + (2^n - K) \cdot (\mu - 1) < K \cdot (T+2) + (2^n - K) \cdot \mu = 2^n \cdot \nu$$

by definition of the expected value threshold ν, and so we are done for this direction.

Having verified both directions of the equivalence given in eq. (21), the correctness of our reduction from the K^{th} largest subset problem is established. Remark that it requires values of the thresholds that are exponential in the size of the set A and polynomial in the value of the largest weight assigned by the size function h (or equivalently, exponential in its encoding) for the K^{th} largest subset problem. It also requires to use edge weights that are polynomial in these values. Observe that this is not a problem, as all those values may be represented using a logarithmic number of bits, hence polynomially in the characteristics of the initial K^{th} largest subset problem. Finally, notice that we do need to consider exponential constants in our game to obtain the NP-hardness of our problem, as for values polynomial in the size of the game, the algorithm described in Sect. 5.1 actually operates in truly-polynomial time. This concludes our proof. □

6 Conclusion

In this paper, we paved the way to a new approach, combining worst-case and expected value requirements in what we named the *beyond worst-case synthesis problem*. We believe this setting is adequate to the synthesis of controllers that must ensure strict guarantees under all circumstances, and prove to be more efficient in reasonable conditions, a problem for which few theoretical frameworks exist.

We thoroughly studied the BWC synthesis problem in the context of two well-known quantitative measures: the *mean-payoff* and the *shortest path*. For the mean-payoff, we proved the problem to be in $\mathrm{NP} \cap \mathrm{coNP}$, matching the complexity of the worst-case threshold problem [30,23,17], which we encompass. Hence, the BWC setting provides additional modeling power at no complexity cost (in terms of problem solving), a surprisingly positive result. For the shortest path, the BWC problem proves to be harder than the worst-case threshold problem, going from polynomial to pseudo-polynomial time, with an NP-hardness result. In both cases, synthesized strategies may require

pseudo-polynomial memory, but accept natural, elegant representations, based on states of the game and simple integer counters.

Possible future works include theoretical study of the BWC problem for other quantitative measures, extension of our results for the mean-payoff and the shortest path to more general settings (multi-dimension [5,8], decidable classes of games with imperfect information [12], etc), and application of the BWC problem to various practical cases.

Acknowledgments. We would like to thank Prof. G. Latouche and G. Louchard for fruitful discussions about Chernoff bounds in Markov models.

References

1. C. Baier and J.-P. Katoen. *Principles of model checking*. MIT Press, 2008.
2. D.P. Bertsekas and J.N. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16:580–595, 1991.
3. L. Brim, J. Chaloupka, L. Doyen, R. Gentilini, and J.-F. Raskin. Faster algorithms for mean-payoff games. *Formal Methods in System Design*, 38(2):97–118, 2011.
4. K. Chatterjee and L. Doyen. Games and Markov decision processes with mean-payoff parity and energy parity objectives. In *Proc. of MEMICS*, LNCS. Springer, 2011.
5. K. Chatterjee, L. Doyen, T.A. Henzinger, and J.-F. Raskin. Generalized mean-payoff and energy games. In *Proc. of FSTTCS*, LIPIcs 8, pages 505–516. Schloss Dagstuhl - LZI, 2010.
6. K. Chatterjee, L. Doyen, M. Randour, and J.-F. Raskin. Looking at mean-payoff and total-payoff through windows. In *Proc. of ATVA*, LNCS 8172, pages 118–132. Springer, 2013.
7. K. Chatterjee and M. Henzinger. An $\mathcal{O}(n^2)$ time algorithm for alternating Büchi games. In *Proc. of SODA*, pages 1386–1399. SIAM, 2012.
8. K. Chatterjee, M. Randour, and J.-F. Raskin. Strategy synthesis for multi-dimensional quantitative objectives. In *Proc. of CONCUR*, LNCS 7454, pages 115–131. Springer, 2012.
9. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995.
10. L. de Alfaro. *Formal verification of probabilistic systems*. PhD thesis, Stanford University, 1997.
11. L. de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In *Proc. of CONCUR*, LNCS 1664, pages 66–81. Springer, 1999.
12. A. Degorre, L. Doyen, R. Gentilini, J.-F. Raskin, and S. Torunczyk. Energy and mean-payoff games with imperfect information. In *Proc. of CSL*, LNCS 6247, pages 260–274. Springer, 2010.
13. A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *Int. Journal of Game Theory*, 8(2):109–113, 1979.
14. J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1997.
15. J.A. Filar, D. Krass, and K.W. Ross. Percentile performance criteria for limiting average Markov decision processes. *Transactions on Automatic Control*, pages 2–10, 1995.
16. M.R. Garey and D.S. Johnson. *Computers and intractability: a guide to the Theory of NP-Completeness*. Freeman New York, 1979.
17. T. Gawlitza and H. Seidl. Games through nested fixpoints. In *Proc. of CAV*, LNCS 5643, pages 291–305. Springer, 2009.
18. H. Gimbert and W. Zielonka. When can you play positionally? In *Proc. of MFCS*, LNCS 3153, pages 686–697. Springer, 2004.
19. P.W. Glynn and D. Ormoneit. Hoeffding's inequality for uniformly ergodic Markov chains. *Statistics & Probability Letters*, 56(2):143–146, 2002.
20. C.M. Grinstead and J.L. Snell. *Introduction to probability*. American Mathematical Society, 1997.
21. W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
22. D.B. Johnson and S.D. Kashdan. Lower bounds for selection in X + Y and other multisets. *Journal of the ACM*, 25(4):556–570, 1978.
23. M. Jurdziński. Deciding the winner in parity games is in UP ∩ co-UP. *Inf. Process. Lett.*, 68(3):119–124, 1998.
24. T.M. Liggett and S.A. Lippman. Stochastic games with perfect information and time average payoff. *Siam Review*, 11(4):604–607, 1969.
25. D.A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975.
26. M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.
27. M. Tracol. Fast convergence to state-action frequency polytopes for MDPs. *Oper. Res. Lett.*, 37(2):123–126, 2009.

28. M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. of FOCS*, pages 327–338. IEEE Computer Society, 1985.
29. C. Wu and Y. Lin. Minimizing risk models in Markov decision processes with policies depending on target values. *Journal of Mathematical Analysis and Applications*, 231(1):47–67, 1999.
30. U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158:343–359, 1996.