Expectation in Stochastic Games with Prefix-independent Objectives

Laurent Doyen
CNRS & LMF, ENS Paris-Saclay, Gif-sur-Yvette, France

Pranshu Gaba
Tata Institute of Fundamental Research, Mumbai, India

Shibashis Guha
Tata Institute of Fundamental Research, Mumbai, India

Abstract

Stochastic two-player games model systems with an environment that is both adversarial and stochastic. In this paper, we study the expected value of bounded quantitative prefix-independent objectives in the context of stochastic games. We show a generic reduction from the expectation problem to linearly many instances of the almost-sure satisfaction problem for threshold Boolean objectives. The result follows from partitioning the vertices of the game into so-called value classes where each class consists of vertices of the same value. Our procedure further entails that the memory required by both players to play optimally for the expectation problem is no more than the memory required by the players to play optimally for the almost-sure satisfaction problem for a corresponding threshold Boolean objective.

We show the applicability of the framework to compute the expected window mean-payoff measure in stochastic games. The window mean-payoff measure strengthens the classical mean-payoff measure by computing the mean payoff over windows of bounded length that slide along an infinite path. We show that the decision problem to check if the expected window mean-payoff value is at least a given threshold is in $\mathsf{UP} \cap \mathsf{coUP}$ when the window length is given in unary.

2012 ACM Subject Classification Probability and statistics → Stochastic processes

Keywords and phrases stochastic games, finitary objectives, mean payoff, reactive synthesis

1 Introduction

Reactive systems typically have an infinite execution where the controller continually reacts to the environment. Given a specification, the *reactive controller synthesis* problem [24] concerns with synthesising a policy for the controller such that the specification is satisfied by the system for all behaviours of the environment. This problem is modelled using two-player turn-based games on graphs, where the two players are the controller (Player 1) and the environment (Player 2), the vertices and the edges of the game graph represent the states and transitions of the system, and the objective of Player 1 is to satisfy the specification. An execution of the system is then an infinite path in the game graph. The reactive controller synthesis problem corresponds to determining if there exists a strategy of Player 1 such that for all strategies of Player 2, the outcome satisfies the objective. If such a winning strategy exists, then we would also like to synthesise it. The environment is considered as an adversarial player to ensure that the specification is met even in the worst-case scenario.

Objectives are either Boolean or quantitative. Each execution either satisfies a Boolean objective ψ or does not satisfy ψ. The set of executions that satisfy ψ form a language over infinite words with the alphabet being the set of vertices in the graphs. On the other hand, a quantitative objective φ evaluates the performance of the execution by a numerical metric, which Player 1 aims to maximise and Player 2 aims to minimise. A quantitative objective can be viewed as a real-valued function over infinite paths in the graph.

In the presence of uncertainty or probabilistic behaviour, the game graph becomes stochastic. Fixing the strategies of the two players gives a distribution over infinite paths in the game graph. For Boolean objectives ψ, the goal of Player 1 is to maximise the probability that an outcome satisfies ψ. For quantitative objectives φ, there are two possible views: *(i) satisfaction*: given a threshold λ, to maximise the probability that φ-value of the outcome is greater than λ; *(ii) expectation*: to maximise the φ-value of the outcome in expectation. Either view may be desirable depending on the context [8,9,11,12]. The satisfaction view can be seen as a Boolean objective: the φ-value of the outcome is either greater than λ or it is not. The expectation view is more nuanced, and is the subject of study in this paper.

In this paper, we look at the expectation problem for quantitative prefix-independent objectives (also referred to as tail objectives). These are objectives that do not depend on finite prefixes of the plays, but only on the long-run behaviour of the system. In systems, we are often willing to allow undesirable behaviour in the short-term, if the long run behaviour is desirable. Prefix-independent objectives model such requirements and thus are of interest to study [13]. Prefix-independent objectives also have the benefit that they satisfy the Bellman equations [39], which simplifies their analysis. The expectation problem for such objectives arises naturally in many scenarios. For example: *(i)* An algorithmic trading system is designed to generate profit by executing trades based on real-time market data. Following an initial phase of learning and unstable behaviour due to parameter tuning, average profit over a bounded time window must always exceed a threshold and decisions need to be made within short well-defined intervals for them to be effective. *(ii)* A power plant may have different strategies to produce power (such as coal, solar, nuclear, wind) and must allocate resources among these strategies so as to maximise the power produced in expectation.

Contributions. All of our contributions are with regard to quantitative prefix-independent objectives φ that are bounded (i.e., the image of φ is bounded between integers $-\mathsf{W}_\varphi$ and W_φ) and such that a bound den_φ on the denominators of the optimal expected φ-values of vertices in the game is known. The bound on the image ensures determinacy [40], that is, the players have optimal strategies, and the bound on the denominator of optimal values of vertices discretise the search space. These bounds often exist and are easily derivable for common objectives of interest such as mean payoff.

Our primary contribution is a reduction of the expectation problem for such an objective φ to linearly many instances of the almost-sure satisfaction problem for threshold Boolean objectives $\{\varphi > \lambda\}$ for thresholds $\lambda \in \mathbb{Q}$. Deciding the almost-sure satisfaction of $\{\varphi > \lambda\}$ is conceptually simpler than computing the expected value of φ, as in the former, we only need to consider if the measure of the paths that satisfy the objective $\{\varphi > \lambda\}$ is equal to one, whereas in the latter, one must take the averages of the measures of the sets of paths π weighted with the value $\varphi(\pi)$ of the paths. Our technique is generic in the sense that when an algorithm for the almost-sure satisfaction problem for $\{\varphi > \lambda\}$ is known, we directly obtain the complexity and a way to solve the expectation problem for φ.

Our reduction builds on the technique introduced in [21] for Boolean prefix-independent objectives and non-trivially extends it to quantitative prefix-independent objectives φ for which the bounds W_φ and den_φ are known. The expected φ-values of vertices are nondeterministically guessed, and we present a characterisation (Theorem 7, similar to [21, Lemma 8] but with important and subtle differences) to verify the guess. We also explicitly construct strategies for both players that are optimal for the expectation of φ, in terms of almost-sure winning strategies for $\{\varphi > \lambda\}$ (proof of Lemma 9). The memory requirement for the constructed optimal strategies is the same as that of the almost-sure winning strategies (Corollary 10).

Our framework gives an alternative approach to solve the expectation problem for well-studied objectives such as expected mean payoff and gives new results for not-as-well-studied objectives such as the *window mean-payoff objectives* introduced in [16]. As our secondary contribution, we illustrate our technique by applying it to two variants of window mean-payoff objectives: fixed ($\mathsf{FWMP}(\ell)$) and bounded (BWMP) window mean-payoff. Using our

Table 1 Complexity and bounds on memory requirement for window mean-payoff objectives

Objective	Complexity	Memory (Player 1) (lower [26], upper)	Memory (Player 2) (lower [26], upper)
$\mathsf{FWMP}(\ell)$	$\mathsf{UP} \cap \mathsf{coUP}$	$\ell - 1$, ℓ	$\lvert V\rvert - \ell$, $\lvert V\rvert \cdot \ell$
BWMP	$\mathsf{UP} \cap \mathsf{coUP}$	memoryless, memoryless	infinite, infinite

reduction, we are able to show that for both of these objectives, the expectation problem is in $\mathsf{UP} \cap \mathsf{coUP}$ (Theorem 18 and Theorem 22), a result that was not known before. The $\mathsf{UP} \cap \mathsf{coUP}$ upper bound for window mean-payoff objectives matches the special case of simple stochastic games [17, 25], and thus would require a major breakthrough to be improved. The lower bounds on the memory requirements for these objectives carry over from special case of the non-stochastic games [16, 26]. We summarise the complexity results and bounds on the memory requirements for the window mean-payoff objectives in Table 1.

Related work. Stochastic games were introduced by Shapley [43] where these games were studied under expectation semantics for discounted-sum objectives. In [18], it was shown that solving stochastic parity games reduces to solving stochastic mean-payoff games. Further, solving stochastic parity games, stochastic mean-payoff games, and simple stochastic games (i.e., stochastic games with reachability objective) are all polynomial-time equivalent [1, 34], and thus, are all in $\mathsf{UP} \cap \mathsf{coUP}$ [17]. A sub-exponential (or even quasi-polynomial) time deterministic algorithms for simple stochastic games on graphs with poly-logarithmic treewidth was proposed in [23]. In [33], sufficient conditions on the objective were shown such that optimal deterministic memoryless strategies exist for the players. In [39], value iteration to solve the expectation problem in stochastic games with reachability, safety, total-payoff, and mean-payoff objectives was studied.

Mean-payoff objectives were studied initially in two-player games, without stochasticity [28, 44], and with stochasticity in [32]. Finitary versions were introduced as window mean-payoff objectives [16]. For finitary mean-payoff objectives, the satisfaction problem [11] and the expectation problem [8] were studied in the special case of Markov decision processes (MDPs), which correspond to stochastic games with a trivial adversary. Expected mean payoff, expected discounted payoff, expected total payoff, etc. are widely studied for MDPs [6, 41]. Both the expectation problem [8] and the satisfaction problem [11] for the $\mathsf{FWMP}(\ell)$ objective are in PTIME, while they are in $\mathsf{UP} \cap \mathsf{coUP}$ for the BWMP objective. Ensuring the satisfaction and expectation semantics simultaneously was studied in MDPs for the mean-payoff objective in [22] and for the window mean-payoff objectives in [30]. In both cases, the complexity was shown to be no greater than that of only expectation optimisation.

The satisfaction problem for window mean-payoff objectives has been studied for two-player stochastic games in [26]. While positive and almost-sure satisfaction of $\mathsf{FWMP}(\ell)$ are in PTIME, it follows from [26] that the problem is in $\mathsf{UP} \cap \mathsf{coUP}$ for quantitative satisfaction i.e., with threshold probabilities $0 < p < 1$. Furthermore, the satisfaction problem of BWMP is in $\mathsf{UP} \cap \mathsf{coUP}$ and thus has the same complexity as that of the special case of MDPs [11].

Due to lack of space, the more intricate proofs and techniques are moved to the appendix while keeping a broad overview in the main body.

2 Preliminaries

Probability distributions. A *probability distribution* over a finite non-empty set A is a function $\mathsf{Pr}\colon A \to [0,1]$ such that $\sum_{a \in A} \mathsf{Pr}(a) = 1$. We denote by $\mathcal{D}(A)$ the set of all probability

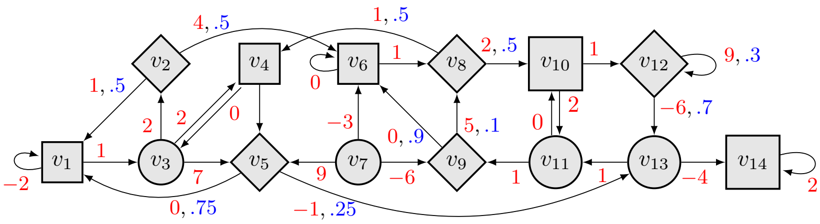

Figure 1 A stochastic game. Player 1 vertices are denoted by circles, Player 2 vertices are denoted by boxes, and probabilistic vertices are denoted by diamonds. The payoff for each edge is shown in red and probability distribution out of probabilistic vertices is shown in blue.

distributions over A. For the algorithmic and complexity results, we assume that probabilities are given as rational numbers.

Stochastic games. We consider two-player turn-based zero-sum stochastic games (or simply, stochastic games in the sequel). The two players are referred to as Player 1 (she/her) and Player 2 (he/him). A *stochastic game* is given by $\mathcal{G} = ((V, E), (V_1, V_2, V_\Diamond), \mathbb{P}, w)$, where:

- (V, E) is a directed graph with a finite set V of vertices and a set $E \subseteq V \times V$ of directed edges such that for all vertices $v \in V$, the set $E(v) = \{v' \in V \mid (v, v') \in E\}$ of out-neighbours of v is nonempty, i.e., $E(v) \neq \varnothing$ (no deadlocks).
- $(V_1, V_2, V_\Diamond)$ is a partition of V. The vertices in V_1 belong to Player 1, the vertices in V_2 belong to Player 2, and the vertices in $V_\Diamond$ are called *probabilistic vertices*;
- $\mathbb{P}\colon V_\Diamond \to \mathcal{D}(V)$ is a *probability function* that describes the behaviour of probabilistic vertices in the game. It maps each probabilistic vertex $v \in V_\Diamond$ to a probability distribution $\mathbb{P}(v)$ over the set $E(v)$ of out-neighbours of v such that $\mathbb{P}(v)(v') > 0$ for all $v' \in E(v)$ (i.e., all out-neighbours have non-zero probability);
- $w\colon E \to \mathbb{Q}$ is a *payoff function* assigning a rational payoff to every edge in the game.

Stochastic games are played in rounds. The game starts by initially placing a token on some vertex. At the beginning of a round, if the token is on a vertex v, and $v \in V_i$ for $i \in \{1, 2\}$, then Player i chooses an out-neighbour $v' \in E(v)$; otherwise $v \in V_\Diamond$, and an out-neighbour $v' \in E(v)$ is chosen with probability $\mathbb{P}(v)(v')$. Player 1 receives from Player 2 the amount $w(v, v')$ given by the payoff function, and the token moves to v' for the next round. This continues ad infinitum resulting in an infinite sequence $\pi = v_0 v_1 v_2 \cdots \in V^\omega$ such that $(v_i, v_{i+1}) \in E$ for all $i \geq 0$, called a *play*. For $i < j$, we denote by $\pi(i, j)$ the *infix* $v_i v_{i+1} \cdots v_j$ of π. Its length is $|\pi(i, j)| = j - i$, the number of edges. We denote by $\pi(0, j)$ the finite *prefix* $v_0 v_1 \cdots v_j$ of π, and by $\pi(i, \infty)$ the infinite *suffix* $v_i v_{i+1} \ldots$ of π. We denote by $\mathsf{Plays}_\mathcal{G}$ and $\mathsf{Prefs}_\mathcal{G}$ the set of all plays and the set of all finite prefixes in $\mathcal{G}$ respectively. We denote by $\mathsf{Last}(\rho)$ the last vertex of the prefix $\rho \in \mathsf{Prefs}_\mathcal{G}$. We denote by $\mathsf{Prefs}^i_\mathcal{G}$ ($i \in \{1, 2\}$) the set of all prefixes ρ such that $\mathsf{Last}(\rho) \in V_i$.

A stochastic game with $V_\Diamond = \varnothing$ is a *non-stochastic two-player game*, a stochastic game with $V_2 = \varnothing$ is a *Markov decision process (MDP)*, and a stochastic game with $V_1 = V_2 = \varnothing$ is a *Markov chain*. Figure 1 shows an example of a stochastic game; Player 1 vertices are shown as circles, Player 2 vertices as boxes, and probabilistic vertices as diamonds.

Subgames and traps. Given a stochastic game $\mathcal{G} = ((V, E), (V_1, V_2, V_\Diamond), \mathbb{P}, w)$, a subset $V' \subseteq V$ of vertices *induces* a subgame if (i) every vertex $v' \in V'$ has an outgoing edge in V', that is $E(v') \cap V' \neq \varnothing$, and (ii) every probabilistic vertex $v' \in V_\Diamond \cap V'$ has all outgoing edges in V', that is $E(v') \subseteq V'$. The induced *subgame* is $((V', E'), (V_1 \cap V', V_2 \cap V', V_\Diamond \cap V'), \mathbb{P}', w')$, where $E' = E \cap (V' \times V')$, and $\mathbb{P}'$ and w' are restrictions of $\mathbb{P}$ and w respectively to (V', E'). If

$T \subseteq V$ is such that for all $v \in T$, if $v \in V_1 \cup V_\Diamond$ then $E(v) \subseteq T$ and if $v \in V_2$ then $E(v) \cap T \neq \varnothing$, then T induces a subgame, and the subgame is a *trap* for Player 1 in $\mathcal{G}$, since Player 2 can ensure that if the token reaches T, then it never escapes.

Boolean objectives. A *Boolean objective* ψ is a Borel-measurable subset of $\mathsf{Plays}_\mathcal{G}$ [40]. A play $\pi \in \mathsf{Plays}_\mathcal{G}$ *satisfies* an objective ψ if $\pi \in \psi$. In a stochastic game $\mathcal{G}$ with objective ψ, the objective of Player 1 is ψ, and since $\mathcal{G}$ is a zero-sum game, the objective of Player 2 is the complement set $\overline{\psi} = \mathsf{Plays}_\mathcal{G} \setminus \psi$. An example of a Boolean objective is *reachability*, denoted $\mathsf{Reach}(T)$, the set of all plays that visit a vertex in the target set $T \subseteq V$. This is formally defined and more examples of Boolean objectives are given in Appendix A.1.

Quantitative objectives. A *quantitative objective* is a Borel-measurable function $\varphi \colon \mathsf{Plays}_\mathcal{G} \to \mathbb{R}$. In a stochastic game $\mathcal{G}$ with objective φ, the objective of Player 1 is φ and the objective of Player 2 is $-\varphi$, the negative of φ. Let $\pi = v_0 v_1 v_2 \cdots$ be a play. Some common examples of quantitative objectives include the *mean-payoff* objective $\mathsf{MP}(\pi) = \liminf_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n} w(v_i, v_{i+1})$, and the *liminf* objective $\mathsf{liminf}(\pi) = \liminf_{n\to\infty} w(v_n, v_{n+1})$. In this work, we also consider the *window mean-payoff* objective, which is defined in Section 4. Corresponding to a quantitative objective φ, we define *threshold objectives* which are Boolean objectives ψ of the form $\{\pi \in \mathsf{Plays}_\mathcal{G} \mid \varphi(\pi) \bowtie \lambda\}$ for thresholds $\lambda \in \mathbb{R}$ and for $\bowtie \in \{<, \leq, >, \geq\}$. We denote this threshold objective succinctly as $\{\varphi \bowtie \lambda\}$.

Prefix independence. An objective is said to be *prefix-independent* if it only depends on the suffix of a play. Formally, a Boolean objective ψ is prefix-independent if for all plays π and π' with a common suffix (that is, π' can be obtained from π by removing and adding a finite prefix), we have that $\pi \in \psi$ if and only if $\pi' \in \psi$. Similarly, a quantitative objective φ is prefix-independent if for all plays π and π' with a common suffix, we have that $\varphi(\pi) = \varphi(\pi')$. Mean payoff and liminf are examples of prefix-independent objectives, whereas reachability and discounted payoff [2] are not.

Strategies. A (deterministic or pure) *strategy*[1] for Player $i \in \{1, 2\}$ in a game $\mathcal{G}$ is a function $\sigma_i : \mathsf{Prefs}^i_\mathcal{G} \to V$ that maps prefixes ending in a vertex $v \in V_i$ to a successor of v. Strategies can be realised as the output of a (possibly infinite-state) Mealy machine [38]. We formally describe the strategy defined by a Mealy machine in Appendix A.1. The *memory size* of a strategy σ_i is the smallest number of states a Mealy machine defining σ_i can have. A strategy σ_i is *memoryless* if $\sigma_i(\rho)$ only depends on the last element of the prefix ρ, that is, for all prefixes $\rho, \rho' \in \mathsf{Prefs}^i_\mathcal{G}$ if $\mathsf{Last}(\rho) = \mathsf{Last}(\rho')$, then $\sigma_i(\rho) = \sigma_i(\rho')$.

A *strategy profile* $\sigma = (\sigma_1, \sigma_2)$ is a pair of strategies σ_1 and σ_2 of Player 1 and Player 2 respectively. A play $\pi = v_0 v_1 \cdots$ is *consistent* with a strategy σ_i of Player i ($i \in \{1, 2\}$) if for all $j \geq 0$ with $v_j \in V_i$, we have $v_{j+1} = \sigma_i(\pi(0, j))$. A play π is an *outcome* of a profile $\sigma = (\sigma_1, \sigma_2)$ if it is consistent with both σ_1 and σ_2. For a Boolean objective ψ, we denote by $\mathsf{Pr}^{\sigma_1,\sigma_2}_{\mathcal{G},v}(\psi)$ the probability that an outcome of the profile (σ_1, σ_2) in $\mathcal{G}$ with initial vertex v satisfies ψ. This is formally defined in Appendix A.1.

Satisfaction probability of Boolean objectives. Let ψ be a Boolean objective. A strategy σ_1 of Player 1 is *winning* with probability p from a vertex v in $\mathcal{G}$ for objective ψ if $\mathsf{Pr}^{\sigma_1,\sigma_2}_{\mathcal{G},v}(\psi) \geq p$ for all strategies σ_2 of Player 2. A strategy σ_1 of Player 1 is *positive* winning (resp., *almost-sure* winning) from v for Player 1 in $\mathcal{G}$ with objective ψ if $\mathsf{Pr}^{\sigma_1,\sigma_2}_{\mathcal{G},v}(\psi) > 0$ (resp., $\mathsf{Pr}^{\sigma_1,\sigma_2}_{\mathcal{G},v}(\psi) = 1$) for all strategies σ_2 of Player 2. In the above, if such a strategy σ_1 exists, then the vertex v is said to be positive winning (resp., almost-sure winning) for Player 1. If a

[1] We only consider the satisfaction and expectation of Borel-measurable objectives, and deterministic strategies suffice for such objectives [14]. Satisfying two goals simultaneously, e.g., $\mathsf{Pr}(\mathsf{Reach}(T_1)) > 0.5 \wedge \mathsf{Pr}(\mathsf{Reach}(T_2)) > 0.5$ requires randomisation and is not allowed by our definition.

vertex v is positive winning (resp., almost-sure winning) for Player 1, then Player 1 is said to play *optimally* from v if she follows a positive (resp., almost-sure) winning strategy from v. We omit analogous definitions for Player 2.

Expected value of quantitative objectives. Let φ be a quantitative objective. Given a strategy profile $\sigma = (\sigma_1, \sigma_2)$ and an initial vertex v, let $\mathbb{E}_v^{\sigma}(\varphi)$ denote the *expected φ-value of the outcome* of the strategy profile σ from v, that is, the expectation of φ over all plays with initial vertex v under the probability measure $\mathsf{Pr}_{\mathcal{G},v}^{\sigma_1,\sigma_2}(\varphi)$. We only consider objectives φ that are Borel-measurable and whose image is bounded. Thus, by determinacy of Blackwell games [40], we have that stochastic games with objective φ are determined. That is, we have $\sup_{\sigma_1} \inf_{\sigma_2} \mathbb{E}_v^{\sigma}(\varphi) = \inf_{\sigma_2} \sup_{\sigma_1} \mathbb{E}_v^{\sigma}(\varphi)$. We call this quantity the *expected φ-value of the vertex v* and denote it by $\mathbb{E}_v(\varphi)$. We say that Player 1 plays *optimally* from a vertex v if she follows a strategy σ_1 such that for all strategies σ_2 of Player 2, the expected φ-value of the outcome is at least $\mathbb{E}_v(\varphi)$. Similarly, Player 2 plays *optimally* if he follows a strategy σ_2 such that for all strategies σ_1 of Player 1, the expected φ-value of the outcome is at most $\mathbb{E}_v(\varphi)$. If φ is a prefix-independent objective, then we have following relation between the expected φ-value of a vertex v and the expected φ-values of its out-neighbours.

▶ **Proposition 1** (*Bellman equations*). *If φ is a prefix-independent objective, then the following equations hold for all $v \in V$.*

$$\mathbb{E}_v(\varphi) = \begin{cases} \max_{v' \in E(v)} \mathbb{E}_{v'}(\varphi) & \textit{if } v \in V_1 \\ \min_{v' \in E(v)} \mathbb{E}_{v'}(\varphi) & \textit{if } v \in V_2 \\ \sum_{v' \in E(v)} \mathbb{P}(v)(v') \cdot \mathbb{E}_{v'}(\varphi) & \textit{if } v \in V_\Diamond \end{cases}$$

In this paper, we consider the expectation problem for prefix-independent objectives. Our solution in turn uses the almost-sure satisfaction problem. The decision problems are defined as follows.

Decision problems. Given a stochastic game $\mathcal{G}$, a quantitative objective φ, a vertex v, and a threshold $\lambda \in \mathbb{Q}$, the following decision problems are relevant:

- *almost-sure satisfaction problem*: Is vertex v almost-sure winning for Player 1 for a threshold objective $\{\varphi > \lambda\}$?
- *expectation problem*: Is $\mathbb{E}_v(\varphi) \geq \lambda$? That is, is the expected φ-value of v at least λ?

The reader is pointed to [2] and [29] for a more comprehensive discussion on the above-mentioned concepts.

3 Reducing expectation to almost-sure satisfaction

In this section, we show a reduction (Theorem 7) of the expectation problem for bounded quantitative prefix-independent objectives φ to the almost-sure satisfaction problem for the corresponding threshold objectives $\{\varphi > \lambda\}$. The reduction involves guessing a value r_v for every vertex v in the game, and then verifying if the guessed values are equal to the expected φ-values of the vertices. Theorem 7 generalises [21, Lemma 8] which studies the satisfaction problem for prefix-independent Boolean objectives, as Boolean objectives can be viewed as a special case of quantitative objectives by restricting the range to $\{0, 1\}$. We further discuss the difference in approaches between Theorem 7 and [21, Lemma 8] in Section 5.

Given a game $\mathcal{G}$ and a bounded prefix-independent quantitative objective φ, our reduction requires the existence of an integer bound den_φ on the denominators of expected φ-values of vertices in $\mathcal{G}$. Since φ is bounded, there exists an integer W_φ such that $|\varphi(\pi)| \leq \mathsf{W}_\varphi$ for every play π in $\mathcal{G}$. Thus, for every vertex v in $\mathcal{G}$, one can write $\mathbb{E}_v(\varphi)$ as $\frac{p}{q}$, where p and q are integers

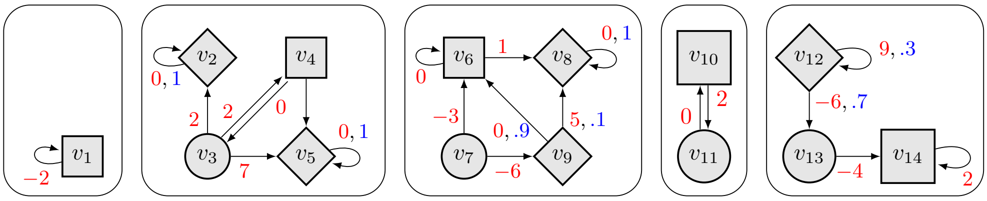

Figure 2 Restrictions $\mathcal{G}_{\mathsf{R}(1)}$, $\mathcal{G}_{\mathsf{R}(2)}$, $\mathcal{G}_{\mathsf{R}(3)}$, $\mathcal{G}_{\mathsf{R}(4)}$, and $\mathcal{G}_{\mathsf{R}(5)}$ of the game shown in Figure 1 for the vector $\vec{\mathsf{r}} = (-2, -1, -1, -1, -1, 0, 0, 0, 0, 1, 1, 2, 2, 2)$.

such that $|p| \leq \mathsf{W}_\varphi \cdot \mathsf{den}_\varphi$ and $0 < q \leq \mathsf{den}_\varphi$. The bounds W_φ and den_φ may depend on the objective and the structure of the graph, i.e., number of vertices, edge payoffs, probability distributions in the game, etc. These bounds effectively discretise the set of possible expected φ-values of the vertices, as there are at most $(2 \cdot \mathsf{W}_\varphi \cdot \mathsf{den}_\varphi + 1) \cdot \mathsf{den}_\varphi$ distinct possible values. This directly gives a bound on the granularity of the possible expected φ-values of vertices, that is, the minimum difference between two possible values of vertices, and we represent this quantity by ε_φ. Observe that given two rational numbers with denominators at most den_φ, the difference between them is at least $(\frac{1}{\mathsf{den}_\varphi})^2$, and thus, we let ε_φ be $(\frac{1}{\mathsf{den}_\varphi})^2$.

3.1 Value vectors and value classes

We first define and give notations for value vectors, which are useful in describing the reduction, and then look at some of their interesting and useful properties.

Definitions and notations. A *vector* $\vec{\mathsf{r}} = (\mathsf{r}_v)_{v \in V}$ of reals indexed by vertices in V induces a partition of V such that all vertices with the same value in $\vec{\mathsf{r}}$ belong to the same part in the partition. Let $k_{\vec{\mathsf{r}}}$ denote the number of parts in the partition, and let us denote the parts by $\{\mathsf{R}(1), \mathsf{R}(2), \ldots, \mathsf{R}(k_{\vec{\mathsf{r}}})\}$. We call each part $\mathsf{R}(i)$ of the partition an $\vec{\mathsf{r}}$-*class*, or simply, *class* if $\vec{\mathsf{r}}$ is clear from the context. For all $1 \leq i \leq k_{\vec{\mathsf{r}}}$, let $\mathfrak{r}_i$ denote the $\vec{\mathsf{r}}$-value of the class $\mathsf{R}(i)$. Given two vectors $\vec{\mathsf{r}}, \vec{\mathsf{s}}$, we write $\vec{\mathsf{r}} \geq \vec{\mathsf{s}}$ if for all $v \in V$, we have $\mathsf{r}_v \geq \mathsf{s}_v$, and we write $\vec{\mathsf{r}} > \vec{\mathsf{s}}$ if we have that $\vec{\mathsf{r}} \geq \vec{\mathsf{s}}$ and there exists $v \in V$ such that $\mathsf{r}_v > \mathsf{s}_v$. For a constant $c \in \mathbb{R}$, we denote by $\vec{\mathsf{r}} + c$ the vector obtained by adding c to each component of $\vec{\mathsf{r}}$.

For all $1 \leq i \leq k_{\vec{\mathsf{r}}}$, a vertex $v \in \mathsf{R}(i)$ is a *boundary vertex* if v is a probabilistic vertex and has an out-neighbour not in $\mathsf{R}(i)$, i.e., if $v \in V_\Diamond$ and $E(v) \not\subseteq \mathsf{R}(i)$. Let $Bnd(\mathsf{R}(i))$ denote the set of boundary vertices in the class $\mathsf{R}(i)$. For all $1 \leq i \leq k_{\vec{\mathsf{r}}}$, let $\mathcal{G}_{\mathsf{R}(i)}$ denote the restriction of $\mathcal{G}$ to vertices in $\mathsf{R}(i)$ with all vertices in $Bnd(\mathsf{R}(i))$ changed to absorbing vertices with a self-loop. The edge payoffs of these self loops are not important (we assume them to be 0) as we restrict our attention to a subgame of $\mathcal{G}_{\mathsf{R}(i)}$ that does not contain boundary vertices.

▶ **Example 2.** For the game shown in Figure 1, let $\vec{\mathsf{r}} = (-2, -1, -1, -1, -1, 0, 0, 0, 0, 1, 1, 2, 2, 2)$ be a value vector for vertices $v_1, v_2, \ldots, v_{14}$ respectively. Since $\vec{\mathsf{r}}$ has five distinct values, we have $k_{\vec{\mathsf{r}}} = 5$, and the five $\vec{\mathsf{r}}$-classes are $\mathsf{R}(1) = \{v_1\}$, $\mathsf{R}(2) = \{v_2, v_3, v_4, v_5\}$, $\mathsf{R}(3) = \{v_6, v_7, v_8, v_9\}$, $\mathsf{R}(4) = \{v_{10}, v_{11}\}$, and $\mathsf{R}(5) = \{v_{12}, v_{13}, v_{14}\}$ with values $\mathfrak{r}_1 = -2$, $\mathfrak{r}_2 = -1$, $\mathfrak{r}_3 = 0$, $\mathfrak{r}_4 = 1$, and $\mathfrak{r}_5 = 2$ respectively. Out of the five probabilistic vertices v_2, v_5, v_8, v_9 and v_{12}, we see that v_2, v_5, and v_8 are boundary vertices while v_9 and v_{12} are not. Thus, $Bnd(\mathsf{R}(2)) = \{v_2, v_5\}$, $Bnd(\mathsf{R}(3)) = \{v_8\}$, and $Bnd(\mathsf{R}(1)) = Bnd(\mathsf{R}(4)) = Bnd(\mathsf{R}(5)) = \varnothing$. We show the restrictions $\mathcal{G}_{\mathsf{R}(i)}$ in Figure 2. ◀

Let φ be a bounded prefix-independent quantitative objective. Analogous to the notation of a general value vector $\vec{\mathsf{r}}$, we describe notations for the expected φ-value vector consisting of the expected φ-values of vertices in V. For all vertices $v \in V$, let s_v denote $\mathbb{E}_v(\varphi)$, the expected φ-value of vertex v, and let $\vec{\mathsf{s}} = (\mathsf{s}_v)_{v \in V}$ denote the expected φ-value vector. Let $\mathsf{S}(i)$ denote the i^{th} $\vec{\mathsf{s}}$-class and let $\mathfrak{s}_i$ denote the $\vec{\mathsf{s}}$-value of $\mathsf{S}(i)$.

Given a vector $\vec{\mathsf{r}}$, it follows from Proposition 1 that the following is a necessary (but not sufficient) condition for $\vec{\mathsf{r}}$ to be the expected φ-value vector $\vec{\mathsf{s}}$.

- ***Bellman*** condition: for every vertex $v \in V$, the following Bellman equations hold

$$\mathsf{r}_v = \begin{cases} \max_{v' \in E(v)} \mathsf{r}_{v'} & \text{if } v \in V_1, \\ \min_{v' \in E(v)} \mathsf{r}_{v'} & \text{if } v \in V_2, \\ \sum_{v' \in E(v)} \mathbb{P}(v)(v') \cdot \mathsf{r}_{v'} & \text{if } v \in V_\Diamond. \end{cases}$$

Consequences of the Bellman condition. We now see some properties of value vectors $\vec{\mathsf{r}}$ that satisfy the **Bellman** condition. Since boundary vertices are probabilistic vertices, the following is immediate.

▶ **Proposition 3.** *Let $\vec{\mathsf{r}}$ be a value vector satisfying the **Bellman** condition. Then for all $1 \le i \le k_{\vec{\mathsf{r}}}$, for all $v \in Bnd(\mathsf{R}(i))$, there exists an out-neighbour of v with $\vec{\mathsf{r}}$-value less than $\mathfrak{r}_i$ and there exists an out-neighbour of v with $\vec{\mathsf{r}}$-value greater than $\mathfrak{r}_i$. Formally, there exist $1 \le i_1, i_2 \le k_{\vec{\mathsf{r}}}$ such that $\mathfrak{r}_{i_1} < \mathfrak{r}_i < \mathfrak{r}_{i_2}$ and $E(v) \cap \mathsf{R}(i_1) \neq \varnothing$ and $E(v) \cap \mathsf{R}(i_2) \neq \varnothing$.*

A corollary of Proposition 3 is that the $\vec{\mathsf{r}}$-classes with the smallest and the biggest $\vec{\mathsf{r}}$-values have no boundary vertices. Note that there may also exist $\vec{\mathsf{r}}$-classes other than these that do not contain boundary vertices (see $\mathsf{R}(4)$ in Example 2). Next, we see that the **Bellman** condition entails that each restriction $\mathcal{G}_{\mathsf{R}(i)}$ is a stochastic game.

▶ **Proposition 4.** *If $\vec{\mathsf{r}}$ is a value vector that satisfies the **Bellman** condition, then for all $1 \le i \le k_{\vec{\mathsf{r}}}$, we have that $\mathcal{G}_{\mathsf{R}(i)}$ is a stochastic game.*

In Proposition 5, we make a crucial observation about long-run behaviours of plays in $\mathcal{G}$, which is that either player can ensure with probability 1 that the token eventually reaches an $\vec{\mathsf{r}}$-class from which it does not exit. This follows from the Borel-Cantelli lemma [27] due to the fact that there is a positive probability to reach an $\vec{\mathsf{r}}$-class without boundary vertices following a finite number of edges out of boundary vertices.

▶ **Proposition 5.** *Let $\vec{\mathsf{r}}$ be a value vector satisfying the **Bellman** condition. Suppose the strategy of Player i ($i \in \{1, 2\}$) is such that each time the token reaches a vertex $v \in V_i$, (s)he moves the token to a vertex v' in the same $\vec{\mathsf{r}}$-class as v. Then, with probability 1, the token eventually reaches a class $\mathsf{R}(j)$ for some $1 \le j \le k_{\vec{\mathsf{r}}}$ from which it never exits.*

Finally, we define the notion of *trap subgames* of $\mathcal{G}_{\mathsf{R}(i)}$ which will be used in the subsequent discussion. We denote by $P^1_{\mathsf{R}(i)}$ the Player 1 positive attractor set of $Bnd(\mathsf{R}(i))$, i.e., the set of vertices in $\mathcal{G}_{\mathsf{R}(i)}$ that are positive winning for Player 1 for the $\mathsf{Reach}(Bnd(\mathsf{R}(i)))$ objective. The complement $T^1_{\mathsf{R}(i)} = \mathsf{R}(i) \setminus P^1_{\mathsf{R}(i)}$ is a trap for Player 1 in $\mathcal{G}_{\mathsf{R}(i)}$, and with abuse of notation, we use the same symbol $T^1_{\mathsf{R}(i)}$ to denote the subset of $\mathcal{G}_{\mathsf{R}(i)}$ as well as the trap subgame. We note that if $\mathsf{R}(i)$ does not have boundary vertices, that is, if $Bnd(\mathsf{R}(i)) = \varnothing$, then it holds that $P^1_{\mathsf{R}(i)} = \varnothing$ and $T^1_{\mathsf{R}(i)} = \mathsf{R}(i)$. We can analogously define $P^2_{\mathsf{R}(i)}$ and $T^2_{\mathsf{R}(i)}$ for Player 2. Given $\mathcal{G}_{\mathsf{R}(i)}$, these sets can be computed in polynomial time using attractor computations [29].

▶ **Example 6.** We compute these sets for the restrictions shown in Figure 2. For $i \in \{1, 4, 5\}$, since $Bnd(\mathsf{R}(i))$ is empty, we have that $T^1_{\mathsf{R}(i)} = \mathsf{R}(i)$ and $P^1_{\mathsf{R}(i)} = \varnothing$. For $\mathsf{R}(2)$, we have that $P^1_{\mathsf{R}(2)} = \mathsf{R}(2)$, and $T^1_{\mathsf{R}(2)} = \varnothing$. For $\mathsf{R}(3)$, we have that $T^1_{\mathsf{R}(3)} = \{v_6\}$, $P^1_{\mathsf{R}(3)} = \{v_7, v_8, v_9\}$. ◀

3.2 Characterisation of the value vector

We describe in Theorem 7 a necessary and sufficient set of conditions for a given vector $\vec{\mathsf{r}}$ to be equal to the expected φ-value vector $\vec{\mathsf{s}}$. In addition to **Bellman**, Theorem 7 makes use of two more conditions, which we define before stating the theorem.

- ***lower-bound*** condition: for all $1 \le i \le k_{\vec{\mathsf{r}}}$, Player 1 wins $\{\varphi > \mathfrak{r}_i - \varepsilon_\varphi\}$ almost surely in the trap subgame $T^1_{\mathsf{R}(i)}$ from all vertices in $T^1_{\mathsf{R}(i)}$.
- ***upper-bound*** condition: for all $1 \le i \le k_{\vec{\mathsf{r}}}$, Player 2 wins $\{\varphi < \mathfrak{r}_i + \varepsilon_\varphi\}$ almost surely in the trap subgame $T^2_{\mathsf{R}(i)}$ from all vertices in $T^2_{\mathsf{R}(i)}$.

▶ **Theorem 7.** *The only vector $\vec{\mathsf{r}}$, whose every component has denominator at most den_φ, that satisfies **Bellman**, **lower-bound**, and **upper-bound** is the expected φ-value vector $\vec{\mathsf{s}}$.*

Proof. We show in Lemma 8 that $\vec{\mathsf{s}}$ satisfies the three conditions. We show in Lemma 9 that if $\vec{\mathsf{r}}$ is a vector that satisfies the three conditions, then $\vec{\mathsf{r}}$ is less than ε_φ distance away from $\vec{\mathsf{s}}$, that is, $\vec{\mathsf{s}} - \varepsilon_\varphi < \vec{\mathsf{r}} < \vec{\mathsf{s}} + \varepsilon_\varphi$. In particular, if each component of $\vec{\mathsf{r}}$ can be written as $\frac{p}{q}$, where p, q are both integers and q is at most den_φ, then it follows that $\vec{\mathsf{r}}$ is equal to $\vec{\mathsf{s}}$. ◀

In the rest of the section, we prove Lemmas 8 and 9 used in the proof of Theorem 7.

▶ **Lemma 8.** *The expected φ-value vector $\vec{\mathsf{s}}$ satisfies the three conditions in Theorem 7.*

Proof. The fact that $\vec{\mathsf{s}}$ satisfies **Bellman** follows directly from Proposition 1. We show that **lower-bound** holds for $\vec{\mathsf{s}}$. The proof for **upper-bound** is analogous.

Suppose for the sake of contradiction that **lower-bound** does not hold, that is, there exists $1 \le i \le k_{\vec{\mathsf{s}}}$ and a vertex v in $T^1_{\mathsf{S}(i)}$ such that Player 2 has a positive winning strategy from v for the $\{\varphi \le \mathfrak{s}_i - \varepsilon_\varphi\}$ objective in $T^1_{\mathsf{S}(i)}$. Since $\{\varphi \le \mathfrak{s}_i - \varepsilon_\varphi\}$ is a prefix-independent objective, from [13, Theorem 1] (restated in Appendix B.2) we have that there exists another vertex v' in $T^1_{\mathsf{S}(i)}$ such that Player 2 has an almost-sure winning strategy from v' for the same objective $\{\varphi \le \mathfrak{s}_i - \varepsilon_\varphi\}$ in $T^1_{\mathsf{S}(i)}$. If Player 2 follows this strategy from v' in the *original game* $\mathcal{G}$, then one of the following two cases holds

- Player 1 always moves the token to a vertex in $\mathsf{S}(i)$. Since v' is in the trap $T^1_{\mathsf{S}(i)}$ for Player 1 in $\mathcal{G}_{\mathsf{S}(i)}$, Player 2 can force the token to remain in $T^1_{\mathsf{S}(i)}$ forever, and follow the almost-sure winning strategy to ensure that with probability 1, the outcome satisfies the objective $\{\varphi \le \mathfrak{s}_i - \varepsilon_\varphi\}$.
- Player 1 eventually moves the token to a vertex out of $\mathsf{S}(i)$. Since $\vec{\mathsf{s}}$ satisfies **Bellman**, the token moves to an $\vec{\mathsf{s}}$-class with a smaller value than $\mathfrak{s}_i$.

In both cases, the expected φ-value of the outcome is less than $\mathfrak{s}_i$. This is a contradiction since $v' \in \mathsf{S}(i)$, and the expected φ-value of every vertex in $\mathsf{S}(i)$ is equal to $\mathfrak{s}_i$. ◀

▶ **Lemma 9.** *If a vector $\vec{\mathsf{r}}$ satisfies the three conditions in Theorem 7, then $\vec{\mathsf{s}} - \varepsilon_\varphi < \vec{\mathsf{r}} < \vec{\mathsf{s}} + \varepsilon_\varphi$. In particular, we have the following:*

- *If $\vec{\mathsf{r}}$ satisfies the **Bellman** and **lower-bound** conditions, then $\vec{\mathsf{s}} > \vec{\mathsf{r}} - \varepsilon_\varphi$.*
- *If $\vec{\mathsf{r}}$ satisfies the **Bellman** and **upper-bound** conditions, then $\vec{\mathsf{s}} < \vec{\mathsf{r}} + \varepsilon_\varphi$.*

Proof sketch. We prove the first case. The proof for the second case follows by symmetry, that is, we essentially replace Player 1 by Player 2, and $\{\varphi > \mathfrak{r}_i - \varepsilon_\varphi\}$ by $\{\varphi < \mathfrak{r}_i + \varepsilon_\varphi\}$. We describe an optimal strategy σ_1^* of Player 1 and give a sketch of its optimality.

Since $\vec{\mathsf{r}}$ satisfies the **lower-bound** condition, we have that for all $1 \le i \le k_{\vec{\mathsf{r}}}$, Player 1 has an an almost-sure winning strategy $\sigma^T_{\mathsf{R}(i)}$ in the trap subgame $T^1_{\mathsf{R}(i)}$ to win the objective

$\{\varphi > \mathfrak{r}_i - \varepsilon_\varphi\}$ in $T^1_{\mathsf{R}(i)}$ almost surely from all vertices in $T^1_{\mathsf{R}(i)}$. From the definition of $P^1_{\mathsf{R}(i)}$, Player 1 has a positive winning strategy $\sigma^P_{\mathsf{R}(i)}$ in the restricted game $\mathcal{G}_{\mathsf{R}(i)}$ from vertices in $P^1_{\mathsf{R}(i)}$ for the $\mathsf{Reach}(Bnd(\mathsf{R}(i)))$ objective. By following $\sigma^P_{\mathsf{R}(i)}$, the token either reaches $Bnd(\mathsf{R}(i))$ with positive probability, or ends up in $T^1_{\mathsf{R}(i)}$ from where Player 2 can ensure that the token never leaves $T^1_{\mathsf{R}(i)}$. Using these strategies of Player 1 in $\mathcal{G}_{\mathsf{R}(i)}$, we construct a strategy σ_1^* of Player 1 that is optimal for expected φ-value in the *original game* $\mathcal{G}$: As long as the token is in the class $\mathsf{R}(i)$ in $\mathcal{G}$, the strategy σ_1^* mimics $\sigma^T_{\mathsf{R}(i)}$ if the token is in $T^1_{\mathsf{R}(i)}$ and σ_1^* mimics $\sigma^P_{\mathsf{R}(i)}$ if the token is in $P^1_{\mathsf{R}(i)}$.

Note that whenever the token is on a vertex $v \in V_1$ in $\mathsf{R}(i)$, the strategy σ_1^* always moves the token to a vertex v' in same $\vec{\mathfrak{r}}$-class $\mathsf{R}(i)$ as v (i.e. a token only exits an $\vec{\mathfrak{r}}$-class from a Player 2 vertex or from a boundary vertex), and thus, Proposition 5 holds. Whenever the token exits a class $\mathsf{R}(i)$ to reach a different class $\mathsf{R}(i')$, then as long as the token remains in $\mathsf{R}(i')$, the strategy σ_1^* follows $\sigma^T_{\mathsf{R}(i')}$ if the token is in $T^1_{\mathsf{R}(i')}$, and σ_1^* follows $\sigma^P_{\mathsf{R}(i')}$ if the token is in $P^1_{\mathsf{R}(i')}$.

Since Proposition 5 holds, we have that for all strategies of Player 2, with probability 1, the token eventually reaches an $\vec{\mathfrak{r}}$-class $\mathsf{R}(j)$ from which it never exits. Moreover, the strategy σ_1^* ensures that with probability 1, the token eventually reaches $T^1_{\mathsf{R}(j)}$ in $\mathsf{R}(j)$ from which it never leaves. Because if not, then the token would visit vertices in $P^1_{\mathsf{R}(j)}$ infinitely often, having a fixed positive probability of reaching $Bnd(\mathsf{R}(j))$ in every step because of $\sigma^P_{\mathsf{R}(j)}$. Thus, with probability 1, the token would eventually reach $Bnd(\mathsf{R}(i))$ from where it could escape to a different $\vec{\mathfrak{r}}$-class, which contradicts the fact that the token stays in $\mathsf{R}(j)$ forever.

Since φ is prefix-independent, the φ-value of a play only depends on the trap $T^1_{\mathsf{R}(j)}$ it ends up in. If the game begins from a vertex $v \in \mathsf{R}(i)$, then for $1 \le j \le k_{\vec{\mathfrak{r}}}$, let p_j denote the probability that the token ends up in the trap subgame $T^1_{\mathsf{R}(j)}$ from which it never exits. Since $\vec{\mathfrak{r}}$ satisfies **Bellman**, we have that $\sum_j p_j \mathfrak{r}_j = \mathfrak{r}_i$. Since $\vec{\mathfrak{r}}$ satisfies **lower-bound**, Player 1 has an almost-sure winning strategy for $\{\varphi > \mathfrak{r}_j - \varepsilon_\varphi\}$ in $T^1_{\mathsf{R}(j)}$. Thus, for all strategies σ_2 of Player 2, the expected value of an outcome of (σ_1^*, σ_2) from $v \in \mathsf{R}(i)$ is greater than $\sum_j p_j(\mathfrak{r}_j - \varepsilon_\varphi)$, which is $\mathfrak{r}_i - \varepsilon_\varphi$. This holds for all vertices v in $\mathcal{G}$, giving us the desired result $\vec{\mathfrak{s}} > \vec{\mathfrak{r}} - \varepsilon_\varphi$. A formal proof is given in Appendix B.5. ◀

We also note that the optimal strategy σ_1^* always either follows an almost-sure winning strategy $\sigma^T_{\mathsf{R}(i)}$ for the threshold objective $\{\varphi > \mathfrak{r}_i - \varepsilon_\varphi\}$ or a positive winning strategy for a Reach objective. Since there exist memoryless positive winning strategies for the Reach objective [25], we have the following bound on the memory requirement of σ_1^*.

▶ **Corollary 10.** *The memory requirement of σ_1^* is at most the maximum over all $1 \le i \le k_{\vec{\mathfrak{r}}}$ of the memory requirement of an almost-sure winning strategy $\sigma^T_{\mathsf{R}(i)}$ for the threshold objective $\{\varphi > \mathfrak{r}_i - \varepsilon_\varphi\}$. Moreover, if $\sigma^T_{\mathsf{R}(i)}$ is a deterministic strategy, then so is σ_1^*.*

3.3 Bounding the denominators in the value vector

In this section, we discuss the problem of obtaining an upper bound den_φ for the denominators of the expected φ-values of vertices $\mathfrak{s}_i$ for a bounded prefix-independent objective φ in a game $\mathcal{G}$. In [21], the technique of value class is used to compute the values of vertices for Boolean prefix-independent objectives. It is stated without proof that the probability of satisfaction of a parity or a Streett objective [2] from each vertex can be written as $\frac{p}{q}$ where $q \le (\widehat{\mathbb{P}})^{4\cdot|E|}$ and $\widehat{\mathbb{P}}$ is the maximum denominator over all edge probabilities in the game. As such, we were not able to directly generalise this bound for the expectation of quantitative prefix-independent objectives. Instead, we make the following observations:

- Let $\mathsf{S}(i)$ be an $\vec{\mathfrak{s}}$-class without boundary vertices. If the token is in $\mathsf{S}(i)$ at some point in the play, then since $\vec{\mathfrak{s}}$ satisfies the **Bellman** condition, neither player has an incentive to move the token out of $\mathsf{S}(i)$. Since there are no boundary vertices in $\mathsf{S}(i)$, the token does not exit $\mathsf{S}(i)$ from a probabilistic vertex either, and remains in $\mathsf{S}(i)$ forever. Thus, the value $\mathfrak{s}_i$ of $\mathsf{S}(i)$ depends only on the internal structure of $\mathsf{S}(i)$. We denote by $\underline{\mathsf{den}}_\varphi$ an upper bound on the denominators of values of $\vec{\mathfrak{s}}$-classes without boundary vertices. It is a simpler problem to find $\underline{\mathsf{den}}_\varphi$ than to find den_φ, as each class without boundary vertices can be treated as a subgame in which each vertex has the same expected φ-value, or equivalently, the subgame consists of exactly one $\vec{\mathfrak{s}}$-class.
- On the other hand, suppose $\mathsf{S}(i)$ is an $\vec{\mathfrak{s}}$-class containing at least one boundary vertex, and let v be a boundary vertex in $\mathsf{S}(i)$. Then, since $\vec{\mathfrak{s}}$ satisfies the **Bellman** condition, we have $\mathsf{s}_v = \sum_{v' \in E(v)} \mathbb{P}(v)(v') \cdot \mathsf{s}_{v'}$, which is also the value $\mathfrak{s}_i$ of $\mathsf{S}(i)$. Thus, $\mathfrak{s}_i$ can be written in terms of the values of classes reachable from v in one step and the probabilities with which those classes are reached. In fact, we construct in the proof of Theorem 11 a system of linear equations to show that the value of each $\vec{\mathfrak{s}}$-class with boundary vertices can be expressed solely in terms of transition probabilities of the outgoing edges from boundary vertices and values of $\vec{\mathfrak{s}}$-classes without boundary vertices.

The method to calculate $\underline{\mathsf{den}}_\varphi$ depends on the specific objective; we illustrate as an example in Section 4 a way to obtain $\underline{\mathsf{den}}_\varphi$ for a particular kind of objective called the window mean-payoff objective. Once we know $\underline{\mathsf{den}}_\varphi$ for an objective φ, we can use Theorem 11 to obtain den_φ in terms of $\underline{\mathsf{den}}_\varphi$.

▶ **Theorem 11.** *The denominator of the value of each $\vec{\mathfrak{s}}$-class in $\mathcal{G}$ is at most $\mathsf{den}_\varphi = 2^{|V|} \cdot \widehat{\mathbb{P}}^{|V|^3} \cdot (\underline{\mathsf{den}}_\varphi)^{|V|}$.*

We note that this theorem implies that the number of bits required to write den_φ is polynomial in the number of vertices in the game and in the number of bits required to write $\underline{\mathsf{den}}_\varphi$. We devote the rest of this section to the proof of Theorem 11. For ease of notation, we denote the number of $\vec{\mathfrak{s}}$-classes in the game by k instead of $k_{\vec{\mathfrak{s}}}$ for the rest of this section. If every $\vec{\mathfrak{s}}$-class has no boundary vertices, then we have den_φ equal to $\underline{\mathsf{den}}_\varphi$ and we are done. So we assume there exists at least one class that contains boundary vertices. Let $m \geq 1$ denote the number of $\vec{\mathfrak{s}}$-classes with boundary vertices, and therefore, there are $k - m$ $\vec{\mathfrak{s}}$-classes without boundary vertices. Since there always exists at least one $\vec{\mathfrak{s}}$-class without boundary vertices (Proposition 3), we have that $m < k$. Let $B = \{1, 2, \ldots, m\}$ and $C = \{m+1, \ldots, k\}$. We index the $\vec{\mathfrak{s}}$-classes such that each class with boundary vertices has its index in B and each class without boundary vertices has its index in C. Furthermore, in the sets B and C, the classes are indexed in increasing order of their values. That is, for i, j both in B or both in C, we have $i < j$ if and only if $\mathfrak{s}_i < \mathfrak{s}_j$. We show bounds on the denominators of $\vec{\mathfrak{s}}$-values of classes with boundary vertices, i.e., $\mathfrak{s}_1, \ldots, \mathfrak{s}_m$ in terms of $\vec{\mathfrak{s}}$-values of classes without boundary vertices, i.e., $\mathfrak{s}_{m+1}, \ldots, \mathfrak{s}_k$.

For all $i \in B = \{1, 2, \ldots, m\}$, pick an arbitrary boundary vertex from $Bnd(\mathsf{S}(i))$ and call this the *representative vertex* u_i of $Bnd(\mathsf{S}(i))$. For all $i \in B$ and $j \in \{1, 2, \ldots, k\}$, let $p_{i,j}$ denote the probability of reaching the class $\mathsf{S}(j)$ from u_i in one step. Since $\vec{\mathfrak{s}}$ satisfies the **Bellman** condition, we have that $\sum_{1 \leq j \leq k} p_{i,j} \cdot \mathfrak{s}_j = \mathfrak{s}_i$. It is helpful to split this sum based on whether $j \in B$ or $j \in C$, i.e., whether $1 \leq j \leq m$ or $m+1 \leq j \leq k$. We rewrite the sums as $\sum_{j \in B} p_{i,j}\mathfrak{s}_j + \sum_{j \in C} p_{i,j}\mathfrak{s}_j = \mathfrak{s}_i$ for all $i \in B$, and we represent this system of equations below using matrices.

$$\begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,m} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m,1} & p_{m,2} & \cdots & p_{m,m} \end{pmatrix} \begin{pmatrix} \mathfrak{s}_1 \\ \mathfrak{s}_2 \\ \vdots \\ \mathfrak{s}_m \end{pmatrix} + \begin{pmatrix} p_{1,m+1} & p_{1,m+2} & \cdots & p_{1,k} \\ p_{2,m+1} & p_{2,m+2} & \cdots & p_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m,m+1} & p_{m,m+2} & \cdots & p_{m,k} \end{pmatrix} \begin{pmatrix} \mathfrak{s}_{m+1} \\ \mathfrak{s}_{m+2} \\ \vdots \\ \mathfrak{s}_k \end{pmatrix} = \begin{pmatrix} \mathfrak{s}_1 \\ \mathfrak{s}_2 \\ \vdots \\ \mathfrak{s}_m \end{pmatrix}$$

This system of equations is of the form $Q_B\mathfrak{s}_B + Q_C\mathfrak{s}_C = \mathfrak{s}_B$. Rearranging terms gives us $(I - Q_B)\mathfrak{s}_B = Q_C\mathfrak{s}_C$ where I is the $m \times m$ identity matrix. It follows from Proposition 12 that the equation $(I - Q_B)\mathfrak{s}_B = Q_C\mathfrak{s}_C$ has a unique solution.

▶ **Proposition 12.** *The matrix $(I - Q_B)$ is invertible.*

Let α denote the least common multiple (lcm) of the denominators of $p_{i,j}$ for $1 \leq i \leq m$ and $1 \leq j \leq k$. We have $0 < \alpha \leq \widehat{\mathbb{P}}^{mk}$, where $\widehat{\mathbb{P}}$ is the maximum denominator over all edge probabilities in $\mathcal{G}$. We multiply both sides of the equation $(I - Q_B)\mathfrak{s}_B = Q_c\mathfrak{s}_C$ by α to get $\alpha(I - Q_B)\mathfrak{s}_B = \alpha Q_C\mathfrak{s}_C$ and note that all the elements of $\alpha(I - Q_B)$ and αQ_C are integers. Let D be the determinant of the matrix $\alpha(I - Q_B)$, and for $1 \leq i \leq m$, let N_i be the determinant of the matrix obtained by replacing the i^{th} column of $\alpha(I - Q_B)$ with the column vector $\alpha Q_C\mathfrak{s}_C$. Since $\alpha(I - Q_B)$ is invertible, by Cramer's rule [37], we have that $\mathfrak{s}_i = N_i/D$. Proposition 13 shows that $|D|$ is an integer and is at most $(2\alpha)^m$ and Proposition 14 shows that N_i has denominator at most $(\underline{\mathsf{den}}_\varphi)^{k-m}$.

▶ **Proposition 13.** *The absolute value of the determinant of $\alpha(I - Q_B)$, i.e., $|D|$, is an integer and is at most $(2\alpha)^m$, which is at most $2^{|V|} \cdot \widehat{\mathbb{P}}^{|V|^3}$.*

▶ **Proposition 14.** *The denominator of N_i is at most $(\underline{\mathsf{den}}_\varphi)^{k-m}$, which is at most $(\underline{\mathsf{den}}_\varphi)^{|V|}$.*

Since the denominator of $\mathfrak{s}_i$ is at most $|D|$ times the denominator of N_i, we obtain the bound stated in Theorem 11. ◀

4 Expectation of window mean-payoff objectives

In this section, we apply the results from the previous section for two types of window mean-payoff objectives introduced in [16]: *(i) fixed window mean-payoff* ($\mathsf{FWMP}(\ell)$) in which a window length $\ell \geq 1$ is given, and *(ii) bounded window mean-payoff* (BWMP) in which for every play, we need a bound on window lengths. We define these objectives below.

For a play π in a stochastic game $\mathcal{G}$, the *mean payoff* of an infix $\pi(i, i+n)$ is the average of the payoffs of the edges in the infix and is defined as $\mathsf{MP}(\pi(i, i+n)) = \sum_{k=i}^{i+n-1} \frac{1}{n} w(v_k, v_{k+1})$. Given a window length $\ell \geq 1$ and a threshold $\lambda \in \mathbb{R}$, a play $\pi = v_0 v_1 \cdots$ in $\mathcal{G}$ satisfies the *fixed window mean-payoff objective* $\mathsf{FWMP}_\mathcal{G}(\ell, \lambda)$ if from every position after some point, it is possible to start an infix of length at most ℓ with mean payoff at least λ.

$$\mathsf{FWMP}_\mathcal{G}(\ell, \lambda) = \{\pi \in \mathsf{Plays}_\mathcal{G} \mid \exists k \geq 0 \cdot \forall i \geq k \cdot \exists j \in \{1, \ldots, \ell\} \colon \mathsf{MP}(\pi(i, i+j)) \geq \lambda\}$$

We omit the subscript $\mathcal{G}$ when it is clear from the context. We extend the definition of *windows* as defined in [16] for arbitrary thresholds. Given a threshold λ, a play $\pi = v_0 v_1 \cdots$, and $0 \leq i < j$, we say that the λ-*window* $\pi(i, j)$ is *open* if the mean payoff of $\pi(i, k)$ is less than λ for all $i < k \leq j$. Otherwise, the λ-window is *closed*. A play π satisfies $\mathsf{FWMP}(\ell, \lambda)$ if and only if from some point on, every λ-window in π closes within at most ℓ steps. Note that $\mathsf{FWMP}(\ell, \lambda) \subseteq \mathsf{FWMP}(\ell', \lambda)$ for $\ell \leq \ell'$ as a smaller window length is a stronger constraint.

We also consider another window mean-payoff objective called the *bounded window mean-payoff objective* $\mathsf{BWMP}_\mathcal{G}(\lambda)$. A play satisfies the objective $\mathsf{BWMP}(\lambda)$ if there exists a window length $\ell \geq 1$ such that the play satisfies $\mathsf{FWMP}(\ell, \lambda)$.

$$\mathsf{BWMP}_\mathcal{G}(\lambda) = \{\pi \in \mathsf{Plays}_\mathcal{G} \mid \exists \ell \geq 1 : \pi \in \mathsf{FWMP}_\mathcal{G}(\ell, \lambda)\}$$

Note that both $\mathsf{FWMP}(\ell, \lambda)$ and $\mathsf{BWMP}(\lambda)$ are Boolean prefix-independent objectives.

Expected window mean-payoff values. Corresponding to the Boolean objectives $\mathsf{FWMP}(\ell, \lambda)$ and $\mathsf{BWMP}(\lambda)$, we define quantitative versions of these objectives. Given a play π in a stochastic game $\mathcal{G}$ and a window length ℓ, the $\varphi_{\mathsf{FWMP}(\ell)}$-value of π is $\sup\{\lambda \in \mathbb{R} \mid \pi \in \mathsf{FWMP}_{\mathcal{G}}(\ell, \lambda)\}$, the supremum threshold λ such that the play satisfies $\mathsf{FWMP}_{\mathcal{G}}(\ell, \lambda)$. Using notations from Section 2, we denote the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of a vertex v by $\mathbb{E}_v(\varphi_{\mathsf{FWMP}(\ell)})$. We define $\mathbb{E}_v(\varphi_{\mathsf{BWMP}})$, the expected φ_{BWMP}-value of a vertex v analogously. If W is an integer such that the payoff $w(e)$ of each edge e in $\mathcal{G}$ satisfies $|w(e)| \leq \mathsf{W}$, then for all plays π in $\mathcal{G}$, we have that $\varphi_{\mathsf{FWMP}(\ell)}(\pi)$ and $\varphi_{\mathsf{BWMP}}(\pi)$ lie between $-\mathsf{W}$ and W. Thus, $\varphi_{\mathsf{FWMP}(\ell)}$ and φ_{BWMP} are bounded objectives.

Decision problems. Given a stochastic game $\mathcal{G}$, a vertex v, and a threshold $\lambda \in \mathbb{Q}$, we have the following expectation problems for the window mean-payoff objectives:

- *expected $\varphi_{\mathsf{FWMP}(\ell)}$-value problem*: Given a window length $\ell \geq 1$, is $\mathbb{E}_v(\varphi_{\mathsf{FWMP}(\ell)}) \geq \lambda$?
- *expected φ_{BWMP}-value problem*: Is $\mathbb{E}_v(\varphi_{\mathsf{BWMP}}) \geq \lambda$?

As considered in previous works [8,11,16], the window length ℓ is usually small ($\ell \leq |V|$), and hence we assume that ℓ is given in unary (while the edge-payoffs are given in binary).

4.1 Expected fixed window mean-payoff value

We give tight complexity bounds for the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value problem. We use the characterisation from Theorem 7 to present our main result that this problem is in $\mathsf{UP} \cap \mathsf{coUP}$ (Theorem 18). We show in Appendix D.1 that simple stochastic games [25], which are known to be in $\mathsf{UP} \cap \mathsf{coUP}$ [17], reduce to the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value problem, giving a tight lower bound.

In order to use the characterisation, we show the existence of the bound $\underline{\mathsf{den}}_{\mathsf{FWMP}(\ell)}$ for the $\varphi_{\mathsf{FWMP}(\ell)}$ objective. We show in Lemma 15 that the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value $\mathfrak{s}_i$ of a class $\mathsf{S}(i)$ without boundary vertices takes a special form, that is, $\mathfrak{s}_i$ is the mean-payoff of a sequence of at most ℓ edges in $\mathsf{S}(i)$.

▶ **Lemma 15.** *The expected $\varphi_{\mathsf{FWMP}(\ell)}$-value $\mathfrak{s}_i$ of vertices in a class $\mathsf{S}(i)$ without boundary vertices is equal to the mean payoff of some sequence of ℓ or fewer edges in $\mathsf{S}(i)$. That is, $\mathfrak{s}_i$ is of the form $\frac{1}{j}\left(w(e_1) + \cdots + w(e_j)\right)$ for some $j \leq \ell$ and edges $e_1, e_2, \ldots, e_j$.*

This observation gives us the bound $\underline{\mathsf{den}}_{\mathsf{FWMP}(\ell)}$ on the denominators of the values of $\vec{\mathfrak{s}}$-classes without boundary vertices. To see this, let $\widehat{w} = \max\{q \mid \exists p, q \in \mathbb{Z}, \exists e \in E : w(e) = \frac{p}{q}$ with p, q co-prime$\}$ be the maximum denominator over all edge-payoffs in $\mathcal{G}$. Since $j \leq \ell$, and each $w(e_1), w(e_2), \ldots, w(e_j)$ is a rational number with denominator at most $\widehat{w}$, the denominator of the sum $w(e_1) + \cdots + w(e_j)$ is at most $\widehat{w} \cdot (\widehat{w} - 1) \cdot (\widehat{w} - 2) \cdots (\widehat{w} - (\ell - 1))$ if $\widehat{w} \geq \ell$, and at most $\widehat{w}!$ if $\widehat{w} \leq \ell$. In both cases, this is at most $\widehat{w}^{\ell}$.

▶ **Corollary 16.** *The expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of vertices in $\vec{\mathfrak{s}}$-classes without boundary vertices can be written as $\frac{p}{q}$ where p and q are integers and $q \leq \widehat{w}^{\ell} \cdot \ell$.*

From Theorem 11, we get that the denominator of $\mathfrak{s}_i$ for each class $\mathsf{S}(i)$ in $\mathcal{G}$ is at most $2^{|V|} \cdot \widehat{\mathbb{P}}^{|V|^3} \cdot (\underline{\mathsf{den}}_{\mathsf{FWMP}(\ell)})^{|V|}$, which is at most $2^{|V|} \cdot \widehat{\mathbb{P}}^{|V|^3} \cdot (\widehat{w}^{\ell} \cdot \ell)^{|V|}$.

▶ **Lemma 17.** *The expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of each vertex in $\mathcal{G}$ can be written as a fraction $\frac{p}{q}$, where p, q are integers, and $q \leq 2^{|V|} \cdot \widehat{\mathbb{P}}^{|V|^3} \cdot (\widehat{w}^{\ell} \cdot \ell)^{|V|}$, and $-\mathsf{W} \cdot q \leq p \leq \mathsf{W} \cdot q$.*

We now state the main result of this section for the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value problem.

▶ **Theorem 18.** *The expected $\varphi_{\mathsf{FWMP}(\ell)}$-value problem is in $\mathsf{UP} \cap \mathsf{coUP}$ when ℓ is given in unary. Memory of size ℓ suffices for Player 1, while memory of size $|V| \cdot \ell$ suffices for Player 2.*

Proof. To show membership of the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value problem in $\mathsf{UP} \cap \mathsf{coUP}$, we first guess the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value vector $\vec{\mathsf{s}}$, that is, the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value s_v of every vertex v in the game. From Lemma 17, it follows that the number of bits required to write s_v for every vertex v is polynomial in the size of the input. Thus, the vector $\vec{\mathsf{s}}$ can be guessed in polynomial time.

Then, to verify the guess, it is sufficient to verify the **Bellman**, **lower-bound**, and **upper-bound** conditions for $\varphi_{\mathsf{FWMP}(\ell)}$. It is easy to see that the **Bellman** condition can be checked in polynomial time. Checking the **lower-bound** and **upper-bound** conditions, i.e., checking the almost-sure satisfaction of the threshold Boolean objective $\mathsf{FWMP}(\ell, \lambda)$ for appropriate thresholds λ in trap subgames in each $\vec{\mathsf{s}}$-class can be done in polynomial time [26]. Thus, the decision problem of $\mathbb{E}_v(\varphi_{\mathsf{FWMP}(\ell)}) \geq \lambda$ is in NP, and moreover, since there is exactly one value vector that satisfies the conditions in Theorem 7, the decision problem is, in fact, in UP. Analogously, the complement decision problem of $\mathbb{E}_v(\varphi_{\mathsf{FWMP}(\ell)}) < \lambda$ is also in UP. Hence, the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value problem is in $\mathsf{UP} \cap \mathsf{coUP}$.

From the description of the optimal strategy in Lemma 9, it follows from Corollary 10 that the memory requirement for the expected $\varphi_{\mathsf{FWMP}(\ell)}$ objective is no greater than the memory requirement for the almost-sure satisfaction of the corresponding threshold objectives, which are ℓ and $|V| \cdot \ell$ for Player 1 and Player 2 respectively [26]. ◀

4.2 Expected bounded window mean-payoff value

We would like to apply the characterisation in Theorem 7 to φ_{BWMP} to show that the expected φ_{BWMP}-value problem is in $\mathsf{UP} \cap \mathsf{coUP}$, and thus, we show the existence of the bound $\mathsf{den}_{\mathsf{BWMP}}$ for the φ_{BWMP} objective. We show in Lemma 19 that the expected φ_{BWMP}-value $\mathfrak{s}_i$ of a class $\mathsf{S}(i)$ without boundary vertices is the mean payoff of a simple cycle in $\mathsf{S}(i)$.

▶ **Lemma 19.** *The expected φ_{BWMP}-value $\mathfrak{s}_i$ of vertices in a class $\mathsf{S}(i)$ without boundary vertices is equal to the mean-payoff value of a simple cycle in $\mathsf{S}(i)$. That is, $\mathfrak{s}_i$ is of the form $\frac{1}{j}(w(e_1) + \cdots + w(e_j))$ for some $j \leq |V|$ and edges $e_1, e_2, \ldots, e_j$ of a simple cycle.*

While Lemma 19 is analogous to Lemma 15 for $\varphi_{\mathsf{FWMP}(\ell)}$, the proof of Lemma 19 is more involved since the φ_{BWMP} objective requires one to consider windows of arbitrary lengths. In the proof, we make use of the fact that memoryless strategies suffice for Player 1 to play optimally for the almost-sure satisfaction of the BWMP objective [26]. In the resulting MDP (which has the same set of vertices as the game $\mathcal{G}_{\mathsf{S}(i)}$), we carefully analyse the resulting plays when Player 2 plays optimally. The formal proof appears in Appendix E.1. The following corollary of Lemma 19 states the bound $\underline{\mathsf{den}}_{\mathsf{BWMP}}$.

▶ **Corollary 20.** *The expected φ_{BWMP}-value of vertices in $\vec{\mathsf{s}}$-classes without boundary vertices can be written as $\frac{p}{q}$ where p and q are integers and $q \leq \widehat{w}^{|V|} \cdot |V|$.*

From Theorem 11, we get that the denominator of $\mathfrak{s}_i$ of each class $\mathsf{S}(i)$ in $\mathcal{G}$ is at most $2^{|V|} \cdot \widehat{\mathbb{P}}^{|V|^3} \cdot (\underline{\mathsf{den}}_{\mathsf{BWMP}})^{|V|}$, which is at most $2^{|V|} \cdot \widehat{\mathbb{P}}^{|V|^3} \cdot (\widehat{w}^{|V|} \cdot |V|)^{|V|}$.

▶ **Lemma 21.** *The expected φ_{BWMP}-value of each vertex in $\mathcal{G}$ can be written as $\frac{p}{q}$, where p, q are integers, and $q \leq 2^{|V|} \cdot \widehat{\mathbb{P}}^{|V|^3} \cdot (\widehat{w}^{|V|} \cdot |V|)^{|V|}$, and $-\mathsf{W} \cdot q \leq p \leq \mathsf{W} \cdot q$.*

We now state the main result of this section for the expected φ_{BWMP}-value problem.

▶ **Theorem 22.** *The expected φ_{BWMP}-value problem is in $\mathsf{UP} \cap \mathsf{coUP}$. Memoryless strategies suffice for Player 1. Player 2 requires infinite memory in general.*

Proof sketch. This proof follows a similar structure as the proof of Theorem 18. As before, the **Bellman** condition can be checked in polynomial time. Checking the **lower-bound** and

upper-bound conditions involves checking almost-sure satisfaction of the Boolean objective BWMP for appropriate thresholds, which reduces to checking the satisfaction of BWMP in non-stochastic games [26], which in turn reduces to total supremum payoff [16], which is in $\mathsf{UP} \cap \mathsf{coUP}$ [31]. Both of these reductions are polynomial-time reduction, and hence, the expected φ_{BWMP}-value problem is in $\mathsf{UP} \cap \mathsf{coUP}$.

Memoryless strategies suffice for Player 1 for almost-sure satisfaction of $\mathsf{BWMP}(\lambda)$ [26]. Player 2 requires infinite memory in general for the $\mathsf{BWMP}(\lambda)$ objective even in non-stochastic games [16], which are a special case of stochastic games. Deterministic strategies suffice for both players. Hence, we get the memory requirements of an optimal strategy for the expected φ_{BWMP}-value problem using Corollary 10. ◀

5 Discussion

We discuss some concluding remarks about the relation of our work to previous work [21], which deals with the satisfaction of Boolean prefix-independent objectives. We also discuss practical implementations for window mean-payoff objectives.

Comparison with [21]. In [21], it suffices to check the almost-sure satisfaction of the same Boolean objective ψ in all value classes. In contrast, for quantitative objectives, the threshold Boolean objective for which we check the almost-sure satisfaction depends on the guessed value of the value class ("Can Player 1 satisfy $\{\varphi > \mathfrak{r}_i - \varepsilon_\varphi\}$ with probability 1?"). Another key difference is that for Boolean objectives, the value classes without boundary vertices are precisely the extremal value classes, that is classes with values 0 and 1. In the case of quantitative objectives, there may be multiple intermediate value classes without boundary vertices, making reasoning about the correctness of the reduction more difficult.

We note that if we apply our approach to Boolean prefix-independent objectives (such as Büchi, coBüchi, parity) by viewing them as quantitative objectives mapping each play to 0 or 1, then we recover the algorithm given in [21].

Practical implementation. We discuss approaches to solve the expected φ-value problem for the window mean-payoff objectives in practice.

A trivial algorithm that works for both $\varphi_{\mathsf{FWMP}(\ell)}$ and φ_{BWMP} objectives is to iterate over all possible value vectors. For each value vector, we check if the conditions in Theorem 7 are satisfied, which can be done in polynomial time. Since there are exponentially many possible value vectors, this algorithm has an exponential running time in the worst-case.

Another technique is value iteration [19], which has been seen to be an anytime algorithm for the standard mean-payoff objective [39]. An anytime algorithm gives better precision the longer it is run, and can be interrupted any time. Given a game $\mathcal{G}$ with $|V|$ vertices, the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value problem on $\mathcal{G}$ reduces to the expected liminf-value problem on a game $\mathcal{G}'$ with $|V|^\ell$ vertices, (that is, on an exponentially larger game graph). The liminf objective is a well-studied objective in the context of value iteration [15, 19]. We describe the reduction in Appendix D.2, which also gives the expected $\varphi_{\mathsf{FWMP}(\ell)}$-values of vertices in $\mathcal{G}$.

Since the size of the graph $\mathcal{G}'$ is much bigger than that of $\mathcal{G}$, we would like to work with $\mathcal{G}'$ on-the-fly rather than explicitly constructing the entire graph. In [39], the authors show bounded value iteration for objectives such as reachability and mean-payoff. They also discuss that the algorithm can be extended to be asynchronous and use partial exploration. As future work, we would like to look at the practicality of on-demand asynchronous value iteration for the liminf objective, or even the window mean-payoff objectives $\varphi_{\mathsf{FWMP}(\ell)}$ and φ_{BWMP} directly. An interesting aspect of it would be to investigate heuristics and optimisations such as sound value iteration [42], optimistic value iteration [36], and topological value iteration [5] to speed up the practical running time.

References

1. D. Andersson and P. B. Miltersen. The Complexity of Solving Stochastic Games on Graphs. In *ISAAC*, volume 5878 of *Lecture Notes in Computer Science*, pages 112–121. Springer, 2009. doi:10.1007/978-3-642-10631-6_13.
2. K. R. Apt and E. Grädel. *Lectures in Game Theory for Computer Scientists*. Cambridge University Press, 2011.
3. P. Ashok, K. Chatterjee, P. Daca, J. Křetínský, and T. Meggendorfer. Value Iteration for Long-Run Average Reward in Markov Decision Processes. In *CAV*, pages 201–221. Springer, 2017. doi:10.1007/978-3-319-63387-9_10.
4. P. Ashok, J. Křetínský, and M. Weininger. PAC Statistical Model Checking for Markov Decision Processes and Stochastic Games. In *CAV*, pages 497–519. Springer, 2019. doi:10.1007/978-3-030-25540-4_29.
5. M. Azeem, A. Evangelidis, J. Křetínský, A. Slivinskiy, and M. Weininger. Optimistic and Topological Value Iteration for Simple Stochastic Games. In *ATVA*, pages 285–302. Springer, 2022. doi:10.1007/978-3-031-19992-9_18.
6. C. Baier and J-P. Katoen. *Principles of model checking*. MIT Press, 2008.
7. P. Billingsley. *Probability and Measure*. John Wiley and Sons, second edition, 1986.
8. B. Bordais, S. Guha, and J-F. Raskin. Expected Window Mean-Payoff. In *FSTTCS*, volume 150 of *LIPIcs*, pages 32:1–32:15, 2019. doi:10.4230/LIPIcs.FSTTCS.2019.32.
9. T. Brázdil, V. Brozek, K. Chatterjee, V. Forejt, and A. Kucera. Two Views on Multiple Mean-Payoff Objectives in Markov Decision Processes. *Logical Methods in Computer Science*, 10(1), 2014. doi:10.23638/LMCS-13(2:15)2017.
10. T. Brázdil, K. Chatterjee, M. Chmelík, V. Forejt, J. Křetínský, M. Kwiatkowska, D. Parker, and M. Ujma. Verification of Markov Decision Processes Using Learning Algorithms. In *ATVA*, pages 98–114. Springer, 2014. doi:10.1007/978-3-319-11936-6_8.
11. T. Brihaye, F. Delgrange, Y. Oualhadj, and M. Randour. Life is Random, Time is Not: Markov Decision Processes with Window Objectives. *Logical Methods in Computer Science*, Volume 16, Issue 4, Dec 2020. doi:10.23638/LMCS-16(4:13)2020.
12. V. Bruyère, E. Filiot, M. Randour, and J-F. Raskin. Meet your expectations with guarantees: Beyond worst-case synthesis in quantitative games. *Information and Computation*, 254:259–295, 2017. doi:10.1016/j.ic.2016.10.011.
13. K. Chatterjee. Concurrent games with tail objectives. *Theoretical Computer Science*, 388(1):181–198, 2007. doi:10.1016/j.tcs.2007.07.047.
14. K. Chatterjee, L. Doyen, H. Gimbert, and T. A. Henzinger. Randomness for free. *Information and Computation*, 245:3–16, 2015. doi:10.1016/j.ic.2015.06.003.
15. K. Chatterjee, L. Doyen, and T. A. Henzinger. A Survey of Stochastic Games with Limsup and Liminf Objectives. In *Automata, Languages and Programming*, pages 1–15. Springer, 2009. doi:10.1007/978-3-642-02930-1_1.
16. K. Chatterjee, L. Doyen, M. Randour, and J-F. Raskin. Looking at mean-payoff and total-payoff through windows. *Information and Computation*, 242:25–52, 2015. doi:10.1016/j.ic.2015.03.010.
17. K. Chatterjee and N. Fijalkow. A reduction from parity games to simple stochastic games. *EPTCS*, 54:74–86, 2011. doi:10.4204/eptcs.54.6.
18. K. Chatterjee and T. A. Henzinger. Reduction of stochastic parity to stochastic mean-payoff games. *Information Processing Letters*, 106(1):1–7, 2008. doi:10.1016/j.ipl.2007.08.035.
19. K. Chatterjee and T. A. Henzinger. Value Iteration. In *25 Years of Model Checking - History, Achievements, Perspectives*, LNCS 5000, pages 107–138. Springer, 2008. doi:10.1007/978-3-540-69850-0_7.
20. K. Chatterjee and T. A. Henzinger. Probabilistic Systems with LimSup and LimInf Objectives. In *Infinity in Logic and Computation*, pages 32–45. Springer, 2009. doi:10.1007/978-3-642-03092-5_4.
21. K. Chatterjee, T. A. Henzinger, and F. Horn. Stochastic Games with Finitary Objectives. In *MFCS*, pages 34–54. Springer, 2009. doi:10.1007/978-3-642-03816-7_4.

22 K. Chatterjee, Z. Křetínská, and J. Křetínský. Unifying Two Views on Multiple Mean-Payoff Objectives in Markov Decision Processes. *Logical Methods in Computer Science*, Volume 13, Issue 2, Jul 2017. doi:10.23638/LMCS-13(2:15)2017.

23 K. Chatterjee, T. Meggendorfer, R. Saona, and J. Svoboda. Faster Algorithm for Turn-based Stochastic Games with Bounded Treewidth. In *SODA*, pages 4590–4605, 2023. doi:10.1137/1.9781611977554.ch173.

24 A. Church. Application of Recursive Arithmetic to the Problem of Circuit Synthesis. *Journal of Symbolic Logic*, 28(4):289–290, 1963. doi:10.2307/2271310.

25 A. Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, 1992. doi:10.1016/0890-5401(92)90048-K.

26 L. Doyen, P. Gaba, and S. Guha. Stochastic Window Mean-payoff Games. In *FoSSaCS Part I*, volume 14574 of *LNCS*, pages 34–54. Springer, 2024. doi:10.1007/978-3-031-57228-9_3.

27 R. Durrett. *Probability: Theory and Examples*. Cambridge University Press, 2010.

28 A. Ehrenfeucht and J. Mycielski. Positional Strategies for Mean Payoff Games. *Int. Journal of Game Theory*, 8(2):109–113, 1979. doi:10.1007/BF01768705.

29 N. Fijalkow, N. Bertrand, P. Bouyer, R. Brenguier, A. Carayol, J. Fearnley, H. Gimbert, F. Horn, R. Ibsen-Jensen, N. Markey, B. Monmege, P. Novotny, M. Randour, O. Sankur, S. Schmitz, O. Serre, and M. Skomra. *Games on Graphs*. 2024. To be published by Cambridge University Press in 2024. doi:10.48550/arXiv.2305.10546.

30 P. Gaba and S. Guha. Optimising expectation with guarantees for window mean payoff in Markov decision processes. In *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems / ACM, to appear, 2025.

31 T. M. Gawlitza and H. Seidl. Games through Nested Fixpoints. In *CAV*, pages 291–305. Springer, 2009. doi:10.1007/978-3-642-02658-4_24.

32 D. Gillette. *Stochastic games with zero stop probabilities*, page 179–188. Princeton University Press, Dec 1958.

33 H. Gimbert and E. Kelmendi. Submixing and Shift-Invariant Stochastic Games. *International Journal of Game Theory*, 52(4):1179–1214, 2023. doi:10.1007/s00182-023-00860-5.

34 V. Gurvich and P. B. Miltersen. On the Computational Complexity of Solving Stochastic Mean-Payoff Games. *CoRR*, abs/0812.0486, 2008.

35 S. Haddad and B. Monmege. Interval Iteration Algorithm for MDPs and IMDPs. *Theoretical Computer Science*, 735:111–131, 2018. Reachability Problems 2014: Special Issue. doi:10.1016/j.tcs.2016.12.003.

36 A. Hartmanns and B. L. Kaminski. Optimistic Value Iteration. In *CAV*, pages 488–511. Springer, 2020. doi:10.1007/978-3-030-53291-8_26.

37 K. Hoffman and R. Kunze. *Linear Algebra*. Prentice-Hall, Inc., 1971.

38 J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Publishing Co., Inc., 1979.

39 J. Křetínský, T. Meggendorfer, and M. Weininger. Stopping Criteria for Value Iteration on Stochastic Games with Quantitative Objectives. In *LICS*, pages 1–14, 2023. doi:10.1109/LICS56636.2023.10175771.

40 D. A. Martin. The Determinacy of Blackwell Games. *The Journal of Symbolic Logic*, 63(4):1565–1581, 1998. doi:10.2307/2586667.

41 M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994.

42 T. Quatmann and J-P. Katoen. Sound Value Iteration. In *CAV*, pages 643–661. Springer, 2018. doi:10.1007/978-3-319-96145-3_37.

43 L. S. Shapley. Stochastic Games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, Oct 1953. doi:10.1073/pnas.39.10.1095.

44 U. Zwick and M. Paterson. The Complexity of Mean Payoff Games on Graphs. *Theoretical Computer Science*, 158(1&2):343–359, 1996. doi:10.1016/0304-3975(95)00188-3.

A Further Preliminaries

A.1 More definitions

One-player games. A stochastic game with $V_1 = V_\Diamond = \varnothing$ or $V_2 = V_\Diamond = \varnothing$ is called a non-stochastic *one-player game.*

Common Boolean objectives. We denote by $\mathrm{occ}(\pi)$ the set of vertices in V that occur at least once in π, and by $\inf(\pi)$ the set of vertices in V that occur infinitely often in π. Given $T \subseteq V$, define the following objectives:

- the *reachability* objective $\mathsf{Reach}_\mathcal{G}(T) = \{\pi \in \mathsf{Plays}_\mathcal{G} \mid T \cap \mathrm{occ}(\pi) \neq \varnothing\}$, the set of all plays that visit a vertex in T,
- the dual *safety* objective $\mathsf{Safe}_\mathcal{G}(T) = \{\pi \in \mathsf{Plays}_\mathcal{G} \mid \mathrm{occ}(\pi) \subseteq T\}$, the set of all plays that never visit a vertex outside T,
- the *Büchi* objective $\mathsf{Büchi}_\mathcal{G}(T) = \{\pi \in \mathsf{Plays}_\mathcal{G} \mid T \cap \inf(\pi) \neq \varnothing\}$, the set of all plays that visit a vertex in T infinitely often, and
- the dual *coBüchi* objective $\mathsf{coBüchi}_\mathcal{G}(T) = \{\pi \in \mathsf{Plays}_\mathcal{G} \mid \inf(\pi) \subseteq T\}$, the set of all plays that eventually only visit vertices in T.

Strategy defined by a Mealy machine. A *Mealy machine* is a deterministic transition system with transitions labelled by input/output pairs. Formally, a Mealy machine M is a tuple $(Q, q_0, \Sigma_i, \Sigma_o, \Delta, \delta)$ where Q is the set of states of M (the memory of the induced strategy), $q_0 \in Q$ is the initial state, Σ_i is the input alphabet, Σ_o is the output alphabet, $\Delta\colon Q \times \Sigma_i \to Q$ is a transition function that reads the current state of M and an input letter and returns the next state of M, and $\delta\colon Q \times \Sigma_i \to \Sigma_o$ is an output function that reads the current state of M and an input letter and returns an output letter. The transition function Δ can be extended to a function $\hat{\Delta}\colon Q \times \Sigma_i^+ \to Q$ that reads words and can be defined inductively by $\hat{\Delta}(q, a) = \Delta(q, a)$ and $\hat{\Delta}(q, x \cdot a) = \Delta(\hat{\Delta}(q, x), a)$, for $q \in Q$, $x \in \Sigma_i^+$, and $a \in \Sigma_i$. The output function δ can be also similarly extended to a function $\hat{\delta}\colon Q \times \Sigma_i^+ \to \Sigma_o$ on words and can be defined inductively by $\hat{\delta}(q, a) = \delta(q, a)$ and $\hat{\delta}(q, x \cdot a) = \delta(\hat{\Delta}(q, x), a)$, for $q \in Q$, $x \in \Sigma_i^+$, and $a \in \Sigma_i$.

A player's strategy can be defined by a Mealy machine whose input and output alphabets are V and $V \cup \{\epsilon\}$ respectively. For $i \in \{1, 2\}$, a strategy σ_i of Player i can be defined by a Mealy machine $(Q, q_0, V, V \cup \{\epsilon\}, \Delta, \delta)$ as follows: Given a prefix $\rho \in \mathsf{Prefs}^i_\mathcal{G}$ ending in a Player i vertex, the strategy σ_i defined by a Mealy machine is $\sigma_i(\rho) = \hat{\delta}(q_0, \rho)$. Intuitively, in each turn, if the token is on a vertex v that belongs to Player i for $i \in \{1, 2\}$, then v is given as input to the Mealy machine, and the Mealy machine outputs the successor vertex of v that Player i must choose. Otherwise, the token is on a vertex v that either belongs to Player i's opponent or is a probabilistic vertex, in which case, the Mealy machine outputs the symbol ϵ to denote that Player i cannot decide the successor vertex of v. Memoryless strategies can be defined by Mealy machines with only one state.

Subgames and subMDPs. Given a stochastic game $\mathcal{G} = ((V, E), (V_1, V_2, V_\Diamond), \mathbb{P}, w)$, a subset $V' \subseteq V$ of vertices *induces* a subgame if (i) every vertex $v' \in V'$ has an outgoing edge in V', that is $E(v') \cap V' \neq \varnothing$, and (ii) every probabilistic vertex $v' \in V_\Diamond \cap V'$ has all outgoing edges in V', that is $E(v') \subseteq V'$. The induced *subgame* is $((V', E'), (V_1 \cap V', V_2 \cap V', V_\Diamond \cap V'), \mathbb{P}', w')$, where $E' = E \cap (V' \times V')$, and $\mathbb{P}'$ and w' are restrictions of $\mathbb{P}$ and w respectively to (V', E'). Let φ be an objective in the stochastic game $\mathcal{G}$. We define the restriction of φ to a subgame $\mathcal{G}'$ of $\mathcal{G}$ to be the set of all plays in $\mathcal{G}'$ satisfying φ, that is, the set $\mathsf{Plays}_{\mathcal{G}'} \cap \varphi$. If $\mathcal{G}$ is an MDP, then a subgame $\mathcal{G}'$ of $\mathcal{G}$ is also an MDP, and is called a *subMDP* of $\mathcal{G}$.

Probability measure over plays. The *cone* at ρ is the set $\mathsf{Cone}(\rho) = \{\pi \in \mathsf{Plays}_{\mathcal{G}} \mid \rho \text{ is a prefix of } \pi\}$, the set of all plays having ρ as a prefix. First, we define this probability measure over cones inductively as follows. If $|\rho| = 0$, then ρ is just a vertex v_0, and $\mathsf{Pr}^{\sigma_1,\sigma_2}_{\mathcal{G},v}(\mathsf{Cone}(\rho))$ is 1 if $v = v_0$, and 0 otherwise. For the inductive case $|\rho| > 0$, there exist $\rho' \in \mathsf{Prefs}_{\mathcal{G}}$ and $v' \in V$ such that $\rho = \rho' \cdot v'$, and we have $\mathsf{Pr}^{\sigma_1,\sigma_2}_{\mathcal{G},v}(\mathsf{Cone}(\rho' \cdot v'))$ given by the following:

$$\begin{cases} \mathsf{Pr}^{\sigma_1,\sigma_2}_{\mathcal{G},v}(\mathsf{Cone}(\rho')) \cdot \mathbb{P}(\mathsf{Last}(\rho'))(v') & \text{if } \mathsf{Last}(\rho') \in V_\Diamond, \\ \mathsf{Pr}^{\sigma_1,\sigma_2}_{\mathcal{G},v}(\mathsf{Cone}(\rho')) & \text{if } \mathsf{Last}(\rho') \in V_i \\ & \text{ and } \sigma_i(\rho') = v', \\ 0 & \text{otherwise.} \end{cases}$$

It is sufficient to define $\mathsf{Pr}^{\sigma_1,\sigma_2}_{\mathcal{G},v}(\varphi)$ on cones in $\mathcal{G}$ since a measure defined on cones extends to a unique measure on $\mathsf{Plays}_{\mathcal{G}}$ by Carathéodory's extension theorem [7].

For a Boolean objective ψ, we denote by $\mathsf{Pr}^{\sigma_1,\sigma_2}_{\mathcal{G},\rho}(\psi)$ the probability that an outcome of the profile (σ_1, σ_2) in $\mathcal{G}$ with initial prefix ρ satisfies ψ. Similarly, we also define $\mathbb{E}^{\sigma}_{\rho}(\varphi)$, the expected φ-value of an outcome of the strategy profile σ following a prefix ρ.

Maximal end components (MECs). Let $\mathcal{G} = ((V, E), (V_1, \varnothing, V_\Diamond), \mathbb{P}, w)$ be an MDP. Given a subset $V' \subseteq V$ of vertices, the subMDP $\mathcal{G}'$ of $\mathcal{G}$ induced by V' is an *end component* of $\mathcal{G}$ if there exists a strategy σ_1 such that for all vertices v, v' in the subgame $\mathcal{G}'$, we have that v' is almost surely reachable from v in $\mathcal{G}'$. Intuitively, if the token is in an end component $\mathcal{G}'$, then Player 1 has a strategy such that, with probability 1, the token never leaves the end component and every vertex is reachable from every other vertex in the end component. An end component that is not contained in any other end component is called a *maximal end component* (MEC).

MECs have traditionally been defined for MDPs only. Recent works such as [4] have extended the definition of MECs to stochastic games in general. Let $\mathcal{G} = ((V, E), (V_1, V_2, V_\Diamond), \mathbb{P}, w)$ be a stochastic game. Given a subset $V' \subseteq V$ of vertices, the subgame $\mathcal{G}'$ of $\mathcal{G}$ induced by V' is an *end component* of $\mathcal{G}$ if there exists a strategy profile (σ_1, σ_2) such that for all vertices v, v' in the subgame $\mathcal{G}'$, we have that v' is almost surely reachable from v in $\mathcal{G}'$. In other words, if the players *cooperate*, then they can ensure that once the token enters an end component, then it never leaves the end component, and that every vertex is reachable from every other vertex in the end component. Equivalently, the subgame $\mathcal{G}'$ of $\mathcal{G}$ induced by V' is an end component of $\mathcal{G}$ if, in the MDP $\mathcal{G}_{\mathsf{MDP}} = ((V, E), (V_1 \cup V_2, \varnothing, V_\Diamond), \mathbb{P}, w)$ obtained by replacing all Player 2 vertices in $\mathcal{G}$ by Player 1 vertices, the subMDP $\mathcal{G}'_{\mathsf{MDP}}$ of $\mathcal{G}_{\mathsf{MDP}}$ induced by V' is an end component of $\mathcal{G}_{\mathsf{MDP}}$ (in the sense of an end component of an MDP).

A.2 Proof of Proposition 1

▶ **Proposition 1** (*Bellman equations*). *If φ is a prefix-independent objective, then the following equations hold for all $v \in V$.*

$$\mathbb{E}_v(\varphi) = \begin{cases} \max_{v' \in E(v)} \mathbb{E}_{v'}(\varphi) & \textit{if } v \in V_1 \\ \min_{v' \in E(v)} \mathbb{E}_{v'}(\varphi) & \textit{if } v \in V_2 \\ \sum_{v' \in E(v)} \mathbb{P}(v)(v') \cdot \mathbb{E}_{v'}(\varphi) & \textit{if } v \in V_\Diamond \end{cases}$$

Proof. First, we show these equations hold for every probabilistic vertex $v \in V_\Diamond$. For every strategy profile σ, we have that $\mathbb{E}^{\sigma}_{v}(\varphi) = \sum_{v' \in E(v)} \mathbb{P}(v)(v') \cdot \mathbb{E}^{\sigma}_{v \cdot v'}(\varphi)$, that is, the expected φ-value of an outcome of the profile σ with initial vertex v is equal to the $\mathbb{P}(v)$-weighted average of the expected φ-value of the outcomes of σ with prefix $v \cdot v'$ for out-neighbours v' of v. Since

φ is prefix-independent, the φ-value of a play is not affected by removing the prefix $v \cdot v'$. Thus, we have that $\mathbb{E}^{\sigma}_{v \cdot v'}(\varphi) = \mathbb{E}^{\sigma}_{v'}(\varphi)$ for all out-neighbours v' of v. Thus, for all strategy profiles σ, we see that $\mathbb{E}^{\sigma}_{v}(\varphi) = \sum_{v' \in E(v)} \mathbb{P}(v)(v') \cdot \mathbb{E}^{\sigma}_{v'}(\varphi)$. In particular, after taking the supremum and the infimum over the strategies of the players, we get $\mathbb{E}_{v}(\varphi) = \sum_{v' \in E(v)} \mathbb{P}(v)(v') \cdot \mathbb{E}_{v'}(\varphi)$.

Now, we show that $\mathbb{E}_v(\varphi) = \max_{v' \in E(v)} \mathbb{E}_{v'}(\varphi)$ for every Player 1 vertex $v \in V_1$. First, we show that $\mathbb{E}_v(\varphi) \geq \max_{v' \in E(v)} \mathbb{E}_{v'}(\varphi)$. Suppose $u \in E(v)$ is an out-neighbour of v with the maximum expected φ-value, that is, $\mathbb{E}_u(\varphi) \geq \mathbb{E}_{v'}(\varphi)$ for all $v' \in E(v)$. Since φ is prefix-independent, for any expected φ-value that can be achieved from u, the same expected φ-value can also be achieved from v by first moving the token to u, and then playing as if the game started from u. Thus, we get that $\mathbb{E}_v(\varphi) \geq \max_{v' \in E(v)} \mathbb{E}_{v'}(\varphi)$. Now, we show the other direction $\mathbb{E}_v(\varphi) \leq \max_{v' \in E(v)} \mathbb{E}_{v'}(\varphi)$. Starting from v, in the most general setting, Player 1 chooses an out-neighbour according to some probability distribution. Since φ is prefix-independent, the expected φ-value of v is a convex combination of the expected φ-values of out-neighbours of v, which is at most $\max_{v' \in E(v)} \mathbb{E}_{v'}(\varphi)$.

We omit the proof for Player 2 vertices $v \in V_2$ as it is analogous to the case of Player 1 vertices. ◀

B Expectation problem: Missing proofs and additional details

B.1 Additional details on trap subgames

▶ **Example 23.** We compute the analogous sets for Player 2 for the restrictions shown in Figure 2. For $i \in \{1, 4, 5\}$, since $Bnd(\mathsf{R}(i))$ is empty, we also have that $T^2_{\mathsf{R}(i)} = \mathsf{R}(i)$ and $P^2_{\mathsf{R}(i)} = \varnothing$. For $i \in \{2, 3\}$, we have that $P^2_{\mathsf{R}(i)} = \mathsf{R}(i)$, and thus, $T^2_{\mathsf{R}(i)} = \varnothing$.

B.2 Statement of [13, Theorem 1]

In a stochastic game with a prefix-independent objective φ, if there exists a vertex that is positive winning, then there exists a vertex v' that is almost-sure winning.

B.3 Proof of Proposition 4

▶ **Proposition 4.** *If $\vec{\mathsf{r}}$ is a value vector that satisfies the **Bellman** condition, then for all $1 \leq i \leq k_{\vec{\mathsf{r}}}$, we have that $\mathcal{G}_{\mathsf{R}(i)}$ is a stochastic game.*

Proof. We need to show that each vertex in $\mathcal{G}_{\mathsf{R}(i)}$ has an out-neighbour in $\mathcal{G}_{\mathsf{R}(i)}$ and the probability distribution over the out-neighbours of probabilistic vertices in $\mathcal{G}_{\mathsf{R}(i)}$ adds up to 1. From the **Bellman** condition, it follows that each non-probabilistic vertex v in $\mathcal{G}_{\mathsf{R}(i)}$ has at least one out-neighbour v' with the same $\vec{\mathsf{r}}$-value as v, and thus v' belongs to $\mathcal{G}_{\mathsf{R}(i)}$. The boundary vertices in $\mathcal{G}_{\mathsf{R}(i)}$ have themself as an out-neighbour. Finally, the out-neighbours of non-boundary probabilistic vertices in $\mathcal{G}_{\mathsf{R}(i)}$ are the same as in $\mathcal{G}$. Hence, $\mathcal{G}_{\mathsf{R}(i)}$ is a stochastic game. ◀

B.4 Proof of Proposition 5

▶ **Proposition 5.** *Let $\vec{\mathsf{r}}$ be a value vector satisfying the **Bellman** condition. Suppose the strategy of Player i ($i \in \{1, 2\}$) is such that each time the token reaches a vertex $v \in V_i$, (s)he moves the token to a vertex v' in the same $\vec{\mathsf{r}}$-class as v. Then, with probability 1, the token eventually reaches a class $\mathsf{R}(j)$ for some $1 \leq j \leq k_{\vec{\mathsf{r}}}$ from which it never exits.*

Proof. We prove this for $i = 1$. The case of $i = 2$ is analogous. If the token exits an $\vec{\mathsf{r}}$-class, it either exits from a boundary vertex or from a Player 2 vertex. From Proposition 3, each time the token reaches a boundary vertex, it has a positive probability of entering a class with a smaller $\vec{r}$-value and a positive probability of entering a class with a greater $\vec{r}$-value. Furthermore, since $\vec{r}$ satisfies the **Bellman** condition, Player 2 can only move the token to a class with a greater $\vec{r}$-value. Since there are $k_{\vec{r}}$ $\vec{r}$-classes, out of which some do not have boundary vertices (in particular the extremal $\vec{r}$-classes do not have boundary vertices), we get that starting from any vertex in the game, it is the case that with probability at least $(\mathbb{P}_{\min})^{k_{\vec{r}}}$, after changing classes at most $k_{\vec{r}}$ times, the token enters a class from which it never exits. Here, $\mathbb{P}_{\min}$ is the minimum probability over all edges in the game. By the second Borel-Cantelli lemma [27], in the infinite play, with probability 1, the token eventually enters a class from which it never exits. ◀

B.5 Proof of optimality of the strategy in Lemma 9

▶ **Lemma 9.** *If a vector $\vec{r}$ satisfies the three conditions in Theorem 7, then $\vec{s} - \varepsilon_\varphi < \vec{r} < \vec{s} + \varepsilon_\varphi$. In particular, we have the following:*

- *If $\vec{r}$ satisfies the **Bellman** and **lower-bound** conditions, then $\vec{s} > \vec{r} - \varepsilon_\varphi$.*
- *If $\vec{r}$ satisfies the **Bellman** and **upper-bound** conditions, then $\vec{s} < \vec{r} + \varepsilon_\varphi$.*

Proof. We formally show that the strategy σ_1^* ensures that for all vertices v in the game $\mathcal{G}$, we have that s_v, the expected φ-value of the outcome in $\mathcal{G}$ starting from v, is greater than $\mathsf{r}_v - \varepsilon_\varphi$. Let σ_2 be an arbitrary strategy[2] of Player 2. Fixing the strategy profile $\sigma = (\sigma_1^*, \sigma_2)$ yields a (possibly infinite) Markov chain $\mathcal{G}_\sigma$, i.e., a probability distribution over the set of plays in $\mathcal{G}$ starting at v consistent with the strategy profile σ. We unfold $\mathcal{G}_\sigma$ to obtain an infinite rooted tree $\mathcal{T}$ with the root being the initial vertex v in $\mathcal{G}$. Each vertex in the tree $\mathcal{T}$ is uniquely identified by the path from the root to that vertex, which corresponds to a unique prefix in the game $\mathcal{G}$. For ease of presentation, we label each vertex of $\mathcal{T}$ by only the last vertex of the prefix. There is a one-to-one correspondence between the set of infinite paths in $\mathcal{T}$ starting at the root v and the set of plays in $\mathcal{G}$ starting from v consistent with σ. For each vertex u in $\mathcal{T}$, if $u \in V_1 \cup V_2$, then u has exactly one child in $\mathcal{T}$ determined by the strategy profile σ, and if $u \in V_\Diamond$, then the probability distribution over the out-neighbours of u in $\mathcal{T}$ is the same as in $\mathcal{G}_\sigma$. Thus, the only branching that occurs in $\mathcal{T}$ is at probabilistic vertices.

We say a vertex u in the tree $\mathcal{T}$ is *final* if all descendants of u in $\mathcal{T}$ (i.e., all vertices reachable from u in $\mathcal{T}$) belong to the same $\vec{r}$-class as u. Once the token reaches a final vertex u in $\mathcal{T}$, i.e., once the token visits the prefix corresponding to u in $\mathcal{G}$, then the token never exits the $\vec{r}$-class that u belongs to. That is, Player 2 never moves the token to a different $\vec{r}$-class and the token never reaches a boundary vertex either. Note that, in particular, every vertex in the $\vec{r}$-class with the greatest value is a final vertex since the class does not have boundary vertices, and all out-neighbours of Player 2 vertices in the class belong to the same class.

We trim final vertices in $\mathcal{T}$, i.e., for each infinite path in $\mathcal{T}$ beginning at the root, we keep the first final vertex u appearing in the path and delete all descendants of u. Let the trimmed tree be denoted by $\hat{\mathcal{T}}$. For $d \geq 1$, let $\hat{\mathcal{T}}_d$ denote the tree $\hat{\mathcal{T}}$ truncated to depth d. That is, for all paths in $\hat{\mathcal{T}}$ of length greater than d, delete all vertices that are at a distance greater than d from the root. Since $\hat{\mathcal{T}}_d$ is finite, starting from the root v of $\hat{\mathcal{T}}_d$, with probability 1, one of the leaves of $\hat{\mathcal{T}}_d$ is reached.

$$\sum_{u \text{ is leaf in } \hat{\mathcal{T}}_d} p_{v \to u} = 1 \tag{1}$$

[2] In this proof, we proceed by assuming that σ_2 is a deterministic strategy. If σ_2 is randomized, then branching will occur not only at vertices in $V_\Diamond$ but also at vertices in V_2 in the tree $\mathcal{T}$. The remaining observations in the proof continue to hold.

Here, $p_{v \to u}$ denotes the probability of reaching u from the root v in $\hat{\mathcal{T}}_d$, that is, the product of the probabilities along the unique path from v to u.

$\vec{\mathsf{r}}$-value of the root in terms of leaves. Let u be a non-final vertex in $\hat{\mathcal{T}}$. We have the following relation between the $\vec{\mathsf{r}}$-values of u and its children in $\hat{\mathcal{T}}$.

- If $u \in V_1$, then $\mathsf{r}_u = \mathsf{r}_{u'}$, where u' is the child of u in $\hat{\mathcal{T}}$. This holds since the strategy σ_1^* of Player 1 always returns a successor vertex in the same $\vec{\mathsf{r}}$-class.
- If $u \in V_2$, then $\mathsf{r}_u \leq \mathsf{r}_{u'}$, where u' is the child of u in $\hat{\mathcal{T}}$. This follows since $\vec{\mathsf{r}}$ satisfies the **Bellman** condition of Theorem 7.
- If $u \in V_\Diamond$, then $\mathsf{r}_u = \sum_{u' \in E(u)} \mathbb{P}(u)(u') \cdot \mathsf{r}_{u'}$. If u is a boundary vertex, then the equation holds since $\vec{\mathsf{r}}$ satisfies the **Bellman** condition of Theorem 7. Otherwise, u is a non-boundary vertex and all out-neighbours of u are in the same $\vec{\mathsf{r}}$-class as u, and thus we have $\mathsf{r}_u = \mathsf{r}_{u'}$ for all out-neighbours u' of u.

By induction on the length of paths from the root v, we get the following relation in $\hat{\mathcal{T}}_d$ for all $d \geq 1$.

$$\mathsf{r}_v \leq \sum_{u \text{ is leaf in } \hat{\mathcal{T}}_d} p_{v \to u} \cdot \mathsf{r}_u \tag{2}$$

$\vec{\mathsf{s}}$-value of the root in terms of leaves. For a vertex u in the tree $\hat{\mathcal{T}}$, let s_u^σ denote the expected φ-value of the outcome of σ following the prefix that is the unique path from the root v to vertex u in $\hat{\mathcal{T}}$. The $\vec{\mathsf{s}}$-value of a non-final vertex u in the tree is equal to the weighted average of the $\vec{\mathsf{s}}$-values of its children in the tree. Formally, we have the following from the Bellman equations in Proposition 1.

- If $u \in V_1 \cup V_2$, then $\mathsf{s}_u^\sigma = \mathsf{s}_{u'}^\sigma$, where u' is the child of u in $\hat{\mathcal{T}}$.
- If $u \in V_\Diamond$, then $\mathsf{s}_u^\sigma = \sum_{u' \in E(u)} \mathbb{P}(u)(u') \cdot \mathsf{s}_{u'}^\sigma$.

By induction on the length of paths from the root v, we get the following relation in $\hat{\mathcal{T}}_d$ for all $d \geq 1$.

$$\mathsf{s}_v^\sigma = \sum_{u \text{ is leaf in } \hat{\mathcal{T}}_d} p_{v \to u} \cdot \mathsf{s}_u^\sigma \tag{3}$$

$\vec{\mathsf{s}}$-values of final vertices. Recall that once the token reaches a final vertex u, the strategy σ_1^* plays the almost-sure winning strategy for the threshold objective $\{\varphi > \mathsf{r}_u - \varepsilon_\varphi\}$. Thus,

$$\text{if } u \text{ is a final vertex, then } \mathsf{s}_u^\sigma > \mathsf{r}_u - \varepsilon_\varphi. \tag{4}$$

Eliminating non-final leaves. If $\hat{\mathcal{T}}$ is finite, then there exists $d \geq 1$ such that we have $\hat{\mathcal{T}} = \hat{\mathcal{T}}_d$. Every leaf of $\hat{\mathcal{T}}_d$ is a final vertex. Thus, from (2), (3), and (4), we get that $\mathsf{s}_v^\sigma > \mathsf{r}_v - \varepsilon_\varphi$ and we are done. Otherwise, suppose $\hat{\mathcal{T}}$ is not finite. Then, for all $d \geq 1$, we have that some of the leaves of $\hat{\mathcal{T}}_d$ are not final. Let $w_{\min}$ denote the minimum φ-value of a play occurring in $\mathcal{G}$. This exists because φ is bounded.

$$\mathsf{s}_u^\sigma \geq w_{\min} \tag{5}$$

We split the sum in (3) depending on whether u is final in $\hat{\mathcal{T}}_d$ or not.

$$\mathsf{s}_v^\sigma = \sum_{\substack{u \text{ is final} \\ \text{in } \hat{\mathcal{T}}_d}} p_{v \to u} \cdot \mathsf{s}_u^\sigma + \sum_{\substack{u \text{ is non-final} \\ \text{leaf in } \hat{\mathcal{T}}_d}} p_{v \to u} \cdot \mathsf{s}_u^\sigma \tag{6}$$

$$> \sum_{\substack{u \text{ is final} \\ \text{in } \hat{\mathcal{T}}_d}} p_{v \to u} \cdot (\mathsf{r}_u - \varepsilon_\varphi) + \sum_{\substack{u \text{ is non-final} \\ \text{leaf in } \hat{\mathcal{T}}_d}} p_{v \to u} \cdot w_{\min} \tag{7}$$

The inequality follows from (4) and (5). Similarly, we can split the sum in (2) depending on whether u is final in $\hat{\mathcal{T}}_d$ or not.

$$\mathsf{r}_v \leq \sum_{\substack{u \text{ is final} \\ \text{in } \hat{\mathcal{T}}_d}} p_{v\to u} \cdot \mathsf{r}_u + \sum_{\substack{u \text{ is non-final} \\ \text{leaf in } \hat{\mathcal{T}}_d}} p_{v\to u} \cdot \mathsf{r}_u \tag{8}$$

Since the strategy σ_1^* never moves the token to a different $\vec{\mathsf{r}}$-class, from Proposition 5, we have that with probability 1, the token eventually reaches a final vertex. From (1), in $\hat{\mathcal{T}}_d$, the probability of reaching a leaf from the root v is 1. Thus, as d increases, the probability measure of final leaves in $\hat{\mathcal{T}}_d$ increases to 1 and the probability of non-final leaves in $\hat{\mathcal{T}}_d$ decreases to 0.

$$\lim_{d\to\infty} \sum_{\substack{u \text{ is final} \\ \text{in } \hat{\mathcal{T}}_d}} p_{v\to u} = 1, \qquad \lim_{d\to\infty} \sum_{\substack{u \text{ is non-final} \\ \text{leaf in } \hat{\mathcal{T}}_d}} p_{v\to u} = 0 \tag{9}$$

The following limits follow from (8) and (9).

$$\lim_{d\to\infty} \sum_{\substack{u \text{ is final} \\ \text{in } \hat{\mathcal{T}}_d}} p_{v\to u} \cdot \mathsf{r}_u \geq \mathsf{r}_v, \quad \lim_{d\to\infty} \sum_{\substack{u \text{ is non-final} \\ \text{leaf in } \hat{\mathcal{T}}_d}} p_{v\to u} \cdot w_{\min} = 0 \tag{10}$$

Thus, from (10), we see that as d increases, the expression in (7) becomes a better approximation of s_v^σ. As $d \to \infty$, we have that $\hat{\mathcal{T}}_d \to \hat{\mathcal{T}}$, and $\mathsf{s}_v^\sigma > \mathsf{r}_v - \varepsilon_\varphi$. Since this holds for arbitrary strategies of Player 2, we get that s_v, the expected φ-value of the initial vertex v, is at least r_v. ◀

B.6 Missing linear algebraic proofs from proof of Theorem 11

▶ Remark 24. The following is the system of equations $(I - Q_B)\mathfrak{s}_B = Q_C\mathfrak{s}_C$ where I is the $m \times m$ identity matrix.

$$\begin{pmatrix} 1-p_{1,1} & -p_{1,2} & \cdots & -p_{1,m} \\ -p_{2,1} & 1-p_{2,2} & \cdots & -p_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ -p_{m,1} & -p_{m,2} & \cdots & 1-p_{m,m} \end{pmatrix} \begin{pmatrix} \mathfrak{s}_1 \\ \mathfrak{s}_2 \\ \vdots \\ \mathfrak{s}_m \end{pmatrix} = \begin{pmatrix} p_{1,m+1} & p_{1,m+2} & \cdots & p_{1,k} \\ p_{2,m+1} & p_{2,m+2} & \cdots & p_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m,m+1} & p_{m,m+2} & \cdots & p_{m,k} \end{pmatrix} \begin{pmatrix} \mathfrak{s}_{m+1} \\ \mathfrak{s}_{m+2} \\ \vdots \\ \mathfrak{s}_k \end{pmatrix}$$

The coefficients $p_{i,j}$ depend on the specific choice of the representative boundary vertices u_i for each $\mathsf{S}(i)$ for $i \in B$. However, since the weighted sum $\sum_{v\in E(u)} \mathbb{P}(u)(v) \cdot \mathsf{s}_v$ is equal for all boundary vertices u belonging to the same class $\mathsf{S}(i)$ by **Bellman**, we have that the solution of $\mathfrak{s}_1, \ldots, \mathfrak{s}_m$ is independent of the chosen representatives. Hence, we are free to choose any representative boundary vertex from each class. ◀

▶ **Proposition 12.** *The matrix $(I - Q_B)$ is invertible.*

Proof. We first show that $\lim_{n\to\infty} Q_B^n = 0$. Then we show how this implies that $I - Q_B$ is invertible.

We show $\lim_{n\to\infty} Q_B^n = 0$ by showing that for every row of Q_B^m (where $m = |B|$), the sum of the elements in the row is strictly less than 1. Recall that for $1 \leq i, j \leq m$, we have that $p_{i,j}$ (the ij^{th} element of Q_B) denotes the probability of reaching $\mathsf{S}(j)$ from u_i in one step, where u_i is the representative boundary vertex chosen from $\mathsf{S}(i)$. The matrix Q_B can be viewed as a transition probability matrix of a Markov chain with states $\{M_1, M_2, \ldots, M_m\}$ (corresponding to the value classes $\mathsf{S}(1), \ldots, \mathsf{S}(m)$ respectively) and an additional sink state (corresponding to the value classes in C). The probability of going from M_i to M_j in one step is $p_{i,j}$ and the probability of going from M_i to the sink state in one step is $1 - \sum_{j=1}^{m} p_{i,j}$.

By Proposition 3, we have that for every $1 \leq i \leq m$, either the sum of the elements in the i^{th} row of Q_B is strictly less than 1, or there exist i_1, i_2 such that $1 \leq i_1 < i < i_2 \leq m$ and $p_{i,i_1} > 0$ and $p_{i,i_2} > 0$. In particular, we have that the sums of elements in the 1^{st} and m^{th} rows are both strictly less than 1.

For $n \geq 1$, the ij^{th} element of Q_B^n denotes the probability of being in state M_j in the Markov chain starting from M_i after exactly n steps. The sum of elements of the i^{th} row of Q_B is the probability that starting from M_i, the Markov chain does not reach the sink state after n steps. Since there are m non-sink states, there is a path of length at most m from each state to the sink state. Thus, the probability that the token reaches the sink state after m steps is positive.

The probability of every transition in the Markov chain is at least $1/\widehat{\mathbb{P}}$. Thus, the probability that the sink state is reached after m steps is at least $1/\widehat{\mathbb{P}}^m$. Thus, the sum of elements in each row of Q_B^m is at most $1 - 1/\widehat{\mathbb{P}}^m$. For all $n \geq 1$, the sum of elements in each row of $(Q_B^m)^n$ is at most $(1 - 1/\widehat{\mathbb{P}}^m)^n$. Thus, as $n \to \infty$, the matrix $Q_B^{mn} \to 0$.

Since $\lim_{n\to\infty} Q_B^n = 0$, we have that the absolute value of every eigenvalue of Q_B is strictly less than 1. To see this, let λ be an eigenvalue of Q_B with eigenvector v. For all $n \geq 0$, we have $Q_B^n v = \lambda^n v$. Since $\lim_{n\to\infty} Q_B^n = 0$, the norm of $Q_B^n v$ goes to 0 as $n \to \infty$. The norm of $\lambda^n v$ also goes to 0 as $n \to \infty$, and hence, we have $|\lambda| < 1$. In particular, 1 is not an eigenvalue of Q_B, and hence, 0 is not an eigenvalue of $I - Q_B$. All eigenvalues of $I - Q_B$ are non-zero, and since the determinant of a matrix is the product of the eigenvalues of the matrix, we have that the determinant of $I - Q_B$ is non-zero. ◀

We show an upper bound for $|D|$, the absolute value of D.

▶ **Proposition 13.** *The absolute value of the determinant of $\alpha(I - Q_B)$, i.e., $|D|$, is an integer and is at most $(2\alpha)^m$, which is at most $2^{|V|} \cdot \widehat{\mathbb{P}}^{|V|^3}$.*

Proof. Every element of $\alpha(I - Q_B)$ is an integer, and hence, D is an integer. To see the upper bound on $|D|$, observe some properties satisfied by $\alpha(I - Q_B)$.

1. Each element in $\alpha(I - Q_B)$ belongs to the set $\{-\alpha, -\alpha + 1, \ldots, 0, \ldots, \alpha - 1, \alpha\}$. This is because every element of $(I - Q_B)$ is between -1 and 1, and because each element of $\alpha(I - Q_B)$ is an integer.
2. For each row of $\alpha(I - Q_B)$, the sum of the absolute values of the elements in the row is at most 2α. To see this, note that the elements in Q_B are between 0 and 1 and each row in Q_B adds up to 1. Thus, the sum of absolute values of elements in a row of $I - Q_B$ is at most 2.

Note that the two properties hold for every minor of $\alpha(I - Q_B)$ as well.

We show that $|D|$ is at most $(2\alpha)^m$ by induction on the minors of $\alpha(I - Q_B)$. Consider any 1×1 minor of $\alpha(I - Q_B)$, that is, an element of $\alpha(I - Q_B)$. The absolute value of such a minor is at most α, which satisfies the property of being at most $(2\alpha)^1$. Now, consider a $t \times t$ submatrix M of $\alpha(I - Q_B)$ for $1 < t \leq m$. By the induction hypothesis, every $(t-1) \times (t-1)$ minor of M has absolute value at most $(2\alpha)^{t-1}$. Pick any row of M and compute the determinant of M by expanding along this row. Since the sum of the absolute values of elements in this row is at most 2α, the absolute value of the determinant of M is at most $(2\alpha) \cdot (2\alpha)^{t-1}$, which is $(2\alpha)^t$. Thus, the absolute value $|D|$ of the determinant D of $\alpha(I - Q_B)$ is at most $(2\alpha)^m$. Since $\alpha \leq \widehat{\mathbb{P}}^{mk}$, $m \leq |V|$, and $k \leq |V|$, we have that $|D|$ is at most $2^{|V|} \cdot \widehat{\mathbb{P}}^{|V|^3}$. ◀

Now, we show an upper bound on the denominator of N_i. Recall that N_i is the determinant of the matrix obtained by replacing the i^{th} column of $\alpha(I - Q_B)$ with $\alpha Q_C \mathfrak{s}_C$.

▶ **Proposition 14.** *The denominator of N_i is at most $(\underline{\mathsf{den}}_\varphi)^{k-m}$, which is at most $(\underline{\mathsf{den}}_\varphi)^{|V|}$.*

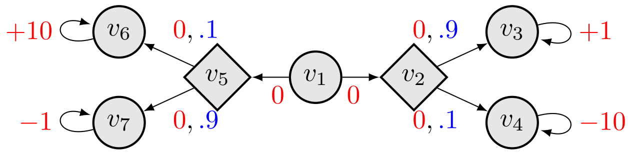

Figure 3 The optimal strategy for Player 1 from v_1 is different in case of maximizing probability of $\mathsf{FWMP}(\ell, 0)$ and maximizing expected $\varphi_{\mathsf{FWMP}(\ell)}$-value.

Proof. Observe that all elements of the matrix αQ_C are integers, and hence, $\alpha Q_C \mathfrak{s}_C$ is a column vector where each element is a weighted sum of $\mathfrak{s}_{m+1}, \mathfrak{s}_{m+2}, \dots, \mathfrak{s}_k$ with integer coefficients. Note that all elements of $\alpha(I - Q_B)$ are integers as well. Thus, N_i is another weighted sum of the form $a_{m+1}\mathfrak{s}_{m+1} + \cdots + a_k \mathfrak{s}_k$ for some integer coefficients $a_{m+1}, a_{m+2}, \dots, a_k$. If written as a fraction in its reduced form, the denominator of N_i is at most the lcm of the denominators of $\mathfrak{s}_{m+1}, \dots, \mathfrak{s}_k$. Since there are at most $(k - m)$ distinct elements in the set of denominators of $\mathfrak{s}_{m+1}, \dots, \mathfrak{s}_k$ and each denominator is at most $\underline{\mathsf{den}}_{\varphi}$, we have that the lcm of this set is at most $(\underline{\mathsf{den}}_{\varphi})^{k-m}$. Thus, the denominator of N_i is at most $(\underline{\mathsf{den}}_{\varphi})^{k-m}$. ◀

C Some properties of window mean-payoff

Inductive property of windows. In [16], the inductive property of λ-windows has been defined for $\lambda = 0$. Here, we generalize this property to arbitrary values of λ.

▶ **Proposition 25** (Inductive property of λ-windows). *If the λ-window starting at position i closes at position j, then for all $i \le k < j$, the λ-windows $\pi(k, j)$ are closed.*

The proof for arbitrary values of λ as compared to value 0 as given in [16] is slightly more involved and hence we present the proof here for completion.

Proof. Since the λ-window starting at position i closes at position j, we have that for all $i \le k < j$, $\mathsf{MP}(\pi(i, k)) < \lambda$ and $\mathsf{MP}(\pi(i, j)) \ge \lambda$. The mean payoff of the infix $\pi(i, j)$ is a weighted average (with positive weights) of the mean payoff of $\pi(i, k)$ and the mean payoff of $\pi(k, j)$. Since $\mathsf{MP}(\pi(i, k)) < \lambda$ and $\mathsf{MP}(\pi(i, j)) \ge \lambda$, this implies that $\mathsf{MP}(\pi(k, j)) \ge \lambda$. Thus, the λ-window starting at position k and ending at position j is closed. ◀

Equivalence of FWMP(1) and Büchi objectives. Note that when $\ell = 1$, the $\mathsf{FWMP}(1, \lambda)$ and $\overline{\mathsf{FWMP}(1, \lambda)}$ (i.e., the complement of $\mathsf{FWMP}(1, \lambda)$) objectives reduce to coBüchi and Büchi objectives respectively. To see this, let T be the set of all vertices $v \in V$ such that either $v \in V_1$ and all out-edges of v have payoff strictly less than λ, or $v \in V_2$ and at least one out-edge of v has a payoff strictly less than λ. Then, a play satisfies the $\overline{\mathsf{FWMP}(1, \lambda)}$ objective if and only if it satisfies the $\mathsf{Büchi}(T)$ objective.

▶ Remark 26. We note that the optimal strategy to maximize the probability of getting nonnegative $\varphi_{\mathsf{FWMP}(\ell)}$-value may be different from the optimal strategy to maximize the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value. Consider the game shown in Figure 3 with initial vertex v_1. If the objective of Player 1 is to maximize the probability of satisfying $\mathsf{FWMP}(\ell, 0)$, then the optimal strategy is to move to v_2 as that ensures a positive $\varphi_{\mathsf{FWMP}(\ell)}$-value with probability 0.9. However, if the optimal strategy is to maximize the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value, then the optimal strategy is to move to v_5. ◀

▶ Remark 27. In an MDP, for all vertices belonging to the same MEC, the expected $\varphi_{\mathsf{FWMP}(\ell)}$-values of all the vertices are the same, and similarly, the expected φ_{BWMP}-values of all the vertices are also the same. However, this is not true for stochastic games in general. Two vertices in the same MEC in a stochastic game may have different expected $\varphi_{\mathsf{FWMP}(\ell)}$-values

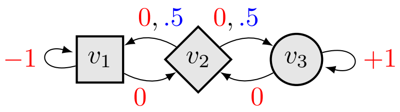

Figure 4 Different vertices in the same MEC in a stochastic game may have different expected $\varphi_{\mathsf{FWMP}(\ell)}$-values.

and different expected φ_{BWMP}-values. Consider the game shown in Figure 4. All three vertices belong to the same MEC since each vertex is reachable from every other vertex. However, the expected $\varphi_{\mathsf{FWMP}(\ell)}$-values of v_1, v_2, and v_3 are -1, 0, and $+1$ respectively, for all values of ℓ. The expected φ_{BWMP}-values are also -1, 0, and $+1$ respectively. ◀

▶ Remark 28 (Achieving supremum for expected $\varphi_{\mathsf{FWMP}(\ell)}$ and φ_{BWMP} objectives.). The $\varphi_{\mathsf{FWMP}(\ell)}$-value of a play is the supremum λ such that every λ-window is closed in at most ℓ steps. Since there are finitely many sequences of edge-payoffs of length at most ℓ, the supremum is achieved for the $\varphi_{\mathsf{FWMP}(\ell)}$ objective for all plays π, and thus can be replaced by max. The supremum for φ_{BWMP}-value may not be reached in general, as Player 2 can keep the window open for increasing window lengths where the supremum is approached asymptotically. ◀

D Expected fixed window mean-payoff value: Missing proofs and additional details

D.1 Reduction to simple stochastic game

▶ **Lemma 29.** *The expected $\varphi_{\mathsf{FWMP}(\ell)}$-value problem is at least as hard as simple stochastic games.*

Proof. The reduction goes as follows. Recall that in a simple stochastic game, Player 1 wins from a vertex v if and only if she has a strategy that ensures that starting from v, the probability that the token eventually reaches the target vertex v_{target} is greater than $\frac{1}{2}$. We assume without loss of generality that v_{target} is absorbing, that is, the only out-neighbour of v_{target} is v_{target} itself.

Given a simple stochastic game $\mathcal{G}_{SSG}$, we construct a new stochastic game $\mathcal{G}$ such that Player 1 reaches v_{target} in $\mathcal{G}_{SSG}$ from v with probability greater than $\frac{1}{2}$ if and only if the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of v in $\mathcal{G}$ is greater than $\frac{1}{2}$.

The set of vertices and edges in $\mathcal{G}$ are the same as in $\mathcal{G}_{SSG}$. We let the edge-payoff of the self-loop of v_{target} be 1 in $\mathcal{G}$, and let the edge-payoff of every other edge in $\mathcal{G}$ be 0. Thus, the probability of reaching v_{target} from v in $\mathcal{G}_{SSG}$ is equal to the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of v in $\mathcal{G}$. Hence, the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value problem is at least as hard as simple stochastic games, which are known to be in $\mathsf{UP} \cap \mathsf{coUP}$. ◀

D.2 $\mathsf{NEXP} \cap \mathsf{coNEXP}$ upper bound

The expected $\varphi_{\mathsf{FWMP}(\ell)}$-value problem can be reduced to the expected liminf-value problem [20] on an exponentially larger game. This gives an $\mathsf{NEXP} \cap \mathsf{coNEXP}$ algorithm for expected $\varphi_{\mathsf{FWMP}(\ell)}$ since the expected liminf-value problem for stochastic games is in $\mathsf{NP} \cap \mathsf{coNP}$ [20].

▶ **Lemma 30.** *The expected $\varphi_{\mathsf{FWMP}(\ell)}$-value problem is in $\mathsf{NEXP} \cap \mathsf{coNEXP}$.*

Proof. Starting with a stochastic game $\mathcal{G} = ((V, E), (V_1, V_2, V_\Diamond), \mathbb{P}, w)$, we construct an exponentially larger stochastic game $\mathcal{G}' = ((V', E'), (V'_1, V'_2, V'_\Diamond), \mathbb{P}', w')$ such that the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of the initial vertex in $\mathcal{G}$ is at least λ if and only if the expected liminf-value of the initial vertex in the constructed game $\mathcal{G}'$ is at least λ.

Intuitively, each vertex in $\mathcal{G}'$ is a history of the last ℓ vertices seen in $\mathcal{G}$, and thus, the game $\mathcal{G}'$ has exponentially many vertices than $\mathcal{G}$. The value of a play in $\mathcal{G}$ depends on the payoffs seen in a sliding window of length ℓ. Since each vertex in $\mathcal{G}'$ stores the ℓ-length history, the value of a play in $\mathcal{G}'$ can be described simply in terms of a liminf objective.

We construct $\mathcal{G}'$ as follows. The set V' of vertices in $\mathcal{G}'$ is equal to $V^{\ell+1}$, and thus, there are $|V|^{\ell+1}$ vertices in $\mathcal{G}'$. We have a label of length $\ell+1$ vertices for each vertex in V'. For all $1 \leq i \leq \ell+1$ and $v' \in V'$, let $L_i(v')$ denote the i^{th} coordinate of v'. That is, $v' = (L_1(v'), L_2(v'), \ldots, L_{\ell+1}(v'))$. Each vertex $v' \in V'$ belongs to V'_1, V'_2, or $V'_\Diamond$ depending on whether the last coordinate of v' (i.e., $L_{\ell+1}(v')$) belongs to V_1, V_2, or $V_\Diamond$. Formally, we have that $v' \in V'_1$ if $L_{\ell+1}(v') \in V_1$, and $v' \in V'_2$ if $L_{\ell+1}(v') \in V_2$, and $v' \in V'_\Diamond$ if $L_{\ell+1}(v') \in V_\Diamond$.

Next, we describe the edges in $\mathcal{G}'$. If v' is a vertex in $\mathcal{G}'$, then for all $u \in V$ such that u is an out-neighbour of $L_{\ell+1}(v')$ in $\mathcal{G}$, we have that $(L_2(v'), L_3(v'), \ldots, L_{\ell+1}(v'), u)$ is an out-neighbour of v' in $\mathcal{G}'$. Formally, $E'(v') = \{(L_2(v'), L_3(v'), \ldots, L_{\ell+1}(v'), u) \mid u \in E(L_{\ell+1}(v'))\}$. Intuitively, we pop out the vertex in the first coordinate and push in a new vertex in the last coordinate. Note that the degree of vertex v' in $\mathcal{G}'$ is equal to the degree of $L_{\ell+1}(v')$ in $\mathcal{G}$.

Now, we define the probability distribution $\mathbb{P}'(v')$ over probabilistic vertices in $\mathcal{G}'$. If v' is a probabilistic vertex, i.e., if $v' \in V'_\Diamond$, then for all out-neighbours $u' \in E'(v')$ of v', we have that $\mathbb{P}'(v')(u') = \mathbb{P}(L_{\ell+1}(v'))(L_{\ell+1}(u'))$.

Finally, we define the payoffs $w'(e')$ of edges e' in $\mathcal{G}'$. All edges coming out of the same vertex in $\mathcal{G}'$ are given the same payoff. That is, the edge-payoffs of (v', u'_1) and (v', u'_2) are equal. The edge-payoff of edges out of v' is determined by the label $L_1(v')L_2(v') \cdots L_{\ell+1}(v')$ of v'. The payoff depends on whether this label is a sequence of edges in $\mathcal{G}$, that is, if $(L_i(v'), L_{i+1}(v')) \in E$ for all $1 \leq i \leq \ell$. For all $v' \in V'$, if the label of v' is a sequence of edges in $\mathcal{G}$, then for all edges e' out of v', i.e., for all $e' \in E'(v')$, the payoff $w'(e')$ of e' is equal to the maximum λ such that the λ-window starting at $L_1(v')$ is closed at or before the end of the label. Otherwise, if it is not a sequence of edges in $\mathcal{G}$, then for all edges e' out of v', we define the payoff of $w'(e')$ to be 0.

If the initial vertex in $\mathcal{G}$ is $v \in V$, then let the initial vertex in $\mathcal{G}'$ be $(v, v, \ldots, v)$, where the tuple has length $\ell+1$ as stated above. Observe that there is a one-to-one correspondence between plays in $\mathcal{G}$ and plays in $\mathcal{G}'$. Starting with a play π in $\mathcal{G}$, we show how to obtain the corresponding play π' in $\mathcal{G}'$. We start with v in π and $(v, v, \ldots, v)$ in π'. Each time we read a vertex, say u, in π, the next vertex in π' can be obtained by considering the label of the last vertex in π' until before u is read in π, popping its first coordinate and pushing the new vertex u in its last coordinate. As a consequence, after the first ℓ steps in $\mathcal{G}'$, every vertex visited in $\mathcal{G}'$ has a label that is a sequence of the last ℓ edges in $\mathcal{G}$. Conversely, given a play π' in $\mathcal{G}'$, one can project π' to the last component to obtain the corresponding play π in $\mathcal{G}$.

If λ is such that, eventually, every λ-window in π closes in at most ℓ steps, then there are infinitely many edges in π' with payoff at least λ, in which case the liminf-value of π' is at least λ. Conversely, if λ is such that there are infinitely many open λ-windows of length ℓ in π, then there are infinitely many edges in π' with payoff less than λ, and the liminf-value of π' is less than λ. The correspondence between plays in the two games, gives a correspondence between strategies between the games. The probability functions $\mathbb{P}$ and $\mathbb{P}'$ of $\mathcal{G}$ and $\mathcal{G}'$ respectively are defined such that once we fix strategies of the players in $\mathcal{G}$, and the corresponding strategies in $\mathcal{G}'$, then the probability distribution of sets of plays in $\mathcal{G}$ is equal to the probability distribution of the corresponding sets of plays in $\mathcal{G}'$. Thus, the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of a vertex v in $\mathcal{G}$ is equal to the expected liminf-value of $(v, v, \ldots, v)$ in $\mathcal{G}'$.

Since the expected liminf-value problem is in $\mathsf{NP} \cap \mathsf{coNP}$ [20] and the size of $\mathcal{G}'$ is exponential in the size of $\mathcal{G}$, we have that the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value problem is in $\mathsf{NEXP} \cap \mathsf{coNEXP}$. ◀

D.3 Proof of Lemma 15

▶ **Lemma 15.** *The expected $\varphi_{\mathsf{FWMP}(\ell)}$-value $\mathfrak{s}_i$ of vertices in a class $\mathsf{S}(i)$ without boundary vertices is equal to the mean payoff of some sequence of ℓ or fewer edges in $\mathsf{S}(i)$. That is, $\mathfrak{s}_i$ is of the form $\frac{1}{j}\left(w(e_1) + \cdots + w(e_j)\right)$ for some $j \leq \ell$ and edges $e_1, e_2, \ldots, e_j$.*

Proof. Let σ_1^* and σ_2^* be optimal strategies of Player 1 and Player 2 respectively for the expected $\varphi_{\mathsf{FWMP}(\ell)}$ objective in $\mathsf{S}(i)$ from an initial vertex v_0 in $\mathsf{S}(i)$. By definition, we have that the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of the outcome of this strategy profile is equal to $\mathfrak{s}_i$. We can in fact show a stronger statement: starting from v_0, with probability 1, the $\varphi_{\mathsf{FWMP}(\ell)}$-value of the outcome of this strategy profile is equal to $\mathfrak{s}_i$. If not, then with positive probability, the $\varphi_{\mathsf{FWMP}(\ell)}$-value of the outcome is greater than $\mathfrak{s}_i$ and with positive probability, the $\varphi_{\mathsf{FWMP}(\ell)}$-value of the outcome is less than $\mathfrak{s}_i$. Since $\varphi_{\mathsf{FWMP}(\ell)}$ is a prefix-independent objective, it follows that there exists another vertex v' from which Player 1 can ensure with probability 1 that the $\varphi_{\mathsf{FWMP}(\ell)}$-value of the outcome is greater than $\mathfrak{s}_i$. It follows that the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of v' is greater than $\mathfrak{s}_i$, which contradicts our hypothesis that the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of each vertex in $\mathsf{S}(i)$ is exactly $\mathfrak{s}_i$.

Now, we show that $\mathfrak{s}_i$ takes the form mentioned in the statement of Lemma 15. Since $\varphi_{\mathsf{FWMP}(\ell)}$ is a prefix-independent objective, the $\varphi_{\mathsf{FWMP}(\ell)}$-value of the outcome π only depends only on $\inf(\pi)$, the set of vertices visited infinitely often in π. For every probabilistic vertex v in $\inf(\pi)$, each out-edge of v is also chosen infinitely often in π. Thus, the $\varphi_{\mathsf{FWMP}(\ell)}$-value of the outcome is equal to supremum λ such that every λ-window closes in at most ℓ steps. This is the mean payoff of a sequence of at most ℓ edges that appears infinitely often in π. This gives that $\mathfrak{s}_i$ is equal to the mean payoff of some j edges in $\mathsf{S}(i)$ for some $1 \leq j \leq \ell$. ◀

D.4 An example illustrating the complexity of computing expected $\varphi_{\mathsf{FWMP}(\ell)}$-value

▶ **Example 31.** We show that the value vector $\vec{\mathfrak{r}}$ in Example 2 for the game in Figure 1 satisfies all the conditions of Theorem 7 for the $\varphi_{\mathsf{FWMP}(\ell)}$ objective with $\ell = 2$. The **Bellman** condition can be verified by analysing the game graph. To check the **lower-bound** condition, we check if Player 1 satisfies the threshold Boolean objectives $\{\varphi_{\mathsf{FWMP}(\ell)} > \mathfrak{r}_i - \varepsilon_{\mathsf{FWMP}}\}$ almost surely from every vertex in each $T^1_{\mathsf{R}(i)}$ (computed in Example 6). Since $T^1_{\mathsf{R}(2)} = \varnothing$, the condition holds vacuously. For $i \in \{1, 3, 5\}$, the game $T^1_{\mathsf{R}(i)}$ consists of only one vertex with a self-loop of value $\mathfrak{r}_i$, and thus the threshold objective is satisfied almost surely. Similarly, for $T^1_{\mathsf{R}(4)}$, there is only one play where payoff 0 and 2 occur alternatingly, and it has $\varphi_{\mathsf{FWMP}(\ell)}$-value equal to 1 for $\ell = 2$. The **upper-bound** condition can be verified analogously. Thus, we have that the vector $\vec{\mathfrak{r}}$ is equal to the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value vector $\vec{\mathfrak{s}}$. ◀

E Expected bounded window mean-payoff value: Additional details

A play π does not satisfy $\mathsf{BWMP}(\lambda)$ if and only if for every suffix of π, for all $\ell \geq 1$, the suffix contains an open λ-window of length ℓ.

▶ Remark 32. In [16], it has been shown that for non-stochastic two-player games, there exists a large enough window length ($\ell_{\max} = (|V| - 1) \cdot (|V| \cdot \mathsf{W} + 1)$, where W is maximum absolute edge payoff in the game) such that for all vertices v in the game, it is the case that v is winning for the $\mathsf{BWMP}(0)$ objective if and only if it is winning for the $\mathsf{FWMP}(\ell_{\max}, 0)$ objective. We remark that in general, there does not exist a window length ℓ such that the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of a vertex is equal to the expected φ_{BWMP}-value of the vertex. To see this, consider the game in Figure 5 (from [16]). We have that the expected φ_{BWMP}-value

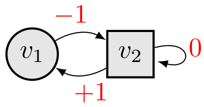

Figure 5 expected φ_{BWMP} value of both vertices is zero, but the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of both vertices is strictly negative for all $\ell \geq 1$.

of both vertices in this game is zero. However, the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of both vertices is equal to $-1/\ell$, which is strictly negative.

E.1 Proof of Lemma 19

▶ **Lemma 19.** *The expected φ_{BWMP}-value $\mathfrak{s}_i$ of vertices in a class $\mathsf{S}(i)$ without boundary vertices is equal to the mean-payoff value of a simple cycle in $\mathsf{S}(i)$. That is, $\mathfrak{s}_i$ is of the form $\frac{1}{j}(w(e_1) + \cdots + w(e_j))$ for some $j \leq |V|$ and edges $e_1, e_2, \ldots, e_j$ of a simple cycle.*

Proof. Since $\mathsf{S}(i)$ is a class without boundary vertices, the game $\mathcal{G}_{\mathsf{S}(i)}$ is obtained by simply restricting $\mathcal{G}$ to the class $\mathsf{S}(i)$. The expected φ_{BWMP}-value of each vertex in $\mathsf{S}(i)$ in the game $\mathcal{G}$ is equal to $\mathfrak{s}_i$, which is also equal to the expected φ_{BWMP}-value of each vertex in the restriction game $\mathcal{G}_{\mathsf{S}(i)}$. Thus, it is sufficient to consider the game $\mathcal{G}_{\mathsf{S}(i)}$. Recall from the proof of Lemma 9 that an optimal strategy of Player 1 for the expected φ_{BWMP}-objective is to follow an optimal strategy for the almost-sure satisfaction of the BWMP objective, for which it is known that memoryless strategies suffice [26]. We show a useful claim that holds for all memoryless strategies σ_1 of Player 1. In particular, the claim also holds for optimal memoryless strategies of Player 1 for which the expected φ_{BWMP}-value $\mathfrak{s}_i$ is attained.

Let σ_1 be a memoryless strategy of Player 1 in $\mathcal{G}_{\mathsf{S}(i)}$. Fixing the strategy σ_1 in the game $\mathcal{G}_{\mathsf{S}(i)}$ gives an MDP $\mathcal{G}^{\sigma_1}_{\mathsf{S}(i)}$ with the same set of vertices as $\mathcal{G}_{\mathsf{S}(i)}$. For ease of notation, we denote this MDP by $\mathcal{M}$. The MDP $\mathcal{M}$ can be decomposed into maximal end-components (MECs). For every MEC T in $\mathcal{M}$, let γ_T denote the mean payoff of the simple cycle in T with the minimum mean payoff. The claim is the following: for every MEC T in $\mathcal{M}$, in the MDP $\mathcal{M}_T$ obtained by restricting $\mathcal{M}$ to T, the expected φ_{BWMP}-value of every vertex is the same and is equal to γ_T. In other words, if the token reaches a MEC T in the MDP $\mathcal{M}$ and Player 2 chooses to always stay in T and never exit, then the expected φ_{BWMP}-value of the outcome is equal to γ_T. We prove this claim later.

Now, we show that this observation implies that $\mathfrak{s}_i$ is equal to the mean payoff of a simple cycle in $\mathsf{S}(i)$. Recall that for all strategies of Player 2, the outcome in $\mathcal{M}$ almost-surely ends up in an MEC of $\mathcal{M}$ from which it never exits. Moreover, since φ_{BWMP} is a prefix-independent objective, the φ_{BWMP}-value of a play only depends on the MEC that the play ends up in. Thus, to obtain the expected φ_{BWMP}-value of vertices in the MDP $\mathcal{M}$, we can *collapse* each MEC M, that is, we replace each MEC M with a single vertex v_M. The out-edges of v_M are the union of the out-edges of vertices in M to vertices not in M, The in-edges of v_M are the union of the in-edges of vertices in M from vertices not in M. In addition, we have a self-loop on v_M with payoff γ_T. The resulting MDP with the collapsed MECs is known as the MEC quotient of $\mathcal{M}$ [3,8,10,35]. The expected φ_{BWMP}-value of vertices in the MDP $\mathcal{M}$ is equal to the expected mean-payoff value of the corresponding vertices in the MDP with the collapsed MECs.

In particular, when the strategy σ_1 of Player 1 is an optimal strategy, we get that every vertex in $\mathcal{M}$ has the same expected φ_{BWMP}-value. That is, for each MEC T in $\mathcal{M}$, we have that γ_T is at least $\mathfrak{s}_i$, and moreover, from each vertex in $\mathcal{M}$, Player 2 has a strategy to almost-surely eventually reach a MEC with value $\mathfrak{s}_i$. Thus, we get that $\mathfrak{s}_i$ is equal to the mean payoff of a simple cycle in $\mathcal{M}$. Since every simple cycle in $\mathcal{M}$ is also a simple cycle in $\mathsf{S}(i)$, we have that $\mathfrak{s}_i$ is equal to the mean payoff of a simple cycle in $\mathsf{S}(i)$.

Proof of claim. It remains to prove our claim. Recall that for a MEC T in the MDP $\mathcal{M}$, we denote by $\mathcal{M}_T$ the restriction of $\mathcal{M}$ to T, and we want to show that for all MECs T in $\mathcal{M}$, the expected φ_{BWMP}-value of every vertex in $\mathcal{M}_T$ is equal to γ_T. First, we show that the expected φ_{BWMP}-value of every vertex in $\mathcal{M}_T$ is at most γ_T. To do this, we show that even if we weaken Player 2, the expected φ_{BWMP}-value of the outcome is at most γ_T. Formally, we weaken Player 2 by replacing every vertex $v \in V_2$ in $\mathcal{M}_T$ that belongs to Player 2 with a probabilistic vertex with a uniform distribution over all out-neighbours of v. This yields a Markov chain which we denote by $\mathcal{C}_T$. Since $\mathcal{M}_T$ is an MEC, we have that $\mathcal{C}_T$ is a single bottom strongly-connected component (BSCC). From [8], it follows that the expected φ_{BWMP}-value of the outcome in a BSCC is equal to the mean payoff of the cycle with the minimum mean payoff in the BSCC. Thus, in the MDP $\mathcal{M}_T$ as well, Player 2 can ensure from every vertex in $\mathcal{M}_T$ that the expected φ_{BWMP}-value of the outcome is at most γ_T.

Now, we show the other direction, that is, the expected φ_{BWMP}-value of every vertex in $\mathcal{M}_T$ is at least γ_T. Here, we strengthen Player 2 by replacing every probabilistic vertex in $\mathcal{M}_T$ with a Player 2 vertex to obtain a non-stochastic one-player game $\mathcal{G}_T$. We show that despite also having control over all probabilistic vertices, for all strategies of Player 2 in $\mathcal{G}_T$, the φ_{BWMP}-value of the outcome in $\mathcal{G}_T$ is at least γ_T. Since $\mathcal{M}_T$ is a MEC, we have that $\mathcal{G}_T$ is strongly connected and that Player 2 has a strategy to reach every vertex in $\mathcal{G}_T$ from every other vertex in $\mathcal{G}_T$. Since φ_{BWMP} is prefix-independent, this implies that the φ_{BWMP}-value of every vertex in $\mathcal{G}_T$ is the same. For ease of analysis, we add $-\gamma_T$ to every edge payoff in $\mathcal{G}_T$ to obtain a new one-player game $\hat{\mathcal{G}}_T$. We have that the φ_{BWMP}-value of vertices in the original game $\mathcal{G}_T$ is at least γ_T if and only if the φ_{BWMP}-value of vertices in the offset game $\hat{\mathcal{G}}_T$ is nonnegative. In the offset game $\hat{\mathcal{G}}_T$, every simple cycle has nonnegative mean payoff, and therefore, every simple cycle in $\hat{\mathcal{G}}_T$ also has nonnegative total payoff. We show that every play in $\hat{\mathcal{G}}_T$ has φ_{BWMP}-value that is nonnegative.

Let $w_{\min}$ denote the minimum edge payoff occurring in the offset game $\hat{\mathcal{G}}_T$. Since there is a simple cycle in $\hat{\mathcal{G}}_T$ with total payoff equal to zero, it cannot be the case that $w_{\min}$ is positive. If $w_{\min}$ is equal to zero, then we have that every edge in $\hat{\mathcal{G}}_T$ has nonnegative payoff, and therefore every play in $\hat{\mathcal{G}}_T$ has φ_{BWMP}-value that is nonnegative. Now, suppose that $w_{\min}$ is strictly negative. Let π be a play in $\hat{\mathcal{G}}_T$. Note that for every suffix $\pi(x,\infty)$ of π, for every finite prefix $\pi(x,y)$ of $\pi(x,\infty)$, the total payoff of the segment $\pi(x,y)$ is bounded below by $|V| \cdot w_{\min}$. Indeed, if the length of $\pi(x,y)$ is strictly less than $|V|$, then the total payoff of the segment is at least $|\pi(x,y)| \cdot w_{\min}$, which is greater than $|V| \cdot w_{\min}$. Otherwise, if the length of $\pi(x,y)$ is at least $|V|$, then it contains a simple cycle with nonnegative total payoff. Deleting this simple cycle from $\pi(x,y)$ gives a shorter play whose total payoff is at most the total payoff of $\pi(x,y)$. Thus, for all $\varepsilon < 0$, we have that all ε-windows in π close in at most $\frac{|V| \cdot w_{\min}}{\varepsilon}$ steps, and thus, the φ_{BWMP}-value of the play π is greater than ε. Thus, the φ_{BWMP}-value of every vertex in the offset game $\hat{\mathcal{G}}_T$ is at least zero, and the φ_{BWMP}-value of every vertex in the one-player game $\mathcal{G}_T$ is at least γ_T. Hence, for all strategies of Player 2 in the MDP $\mathcal{M}_T$, the expected φ_{BWMP}-value of the outcome in $\mathcal{M}_T$ is also at least γ_T.

This shows that the expected φ_{BWMP}-value of every vertex in $\mathcal{M}_T$ is equal to γ_T and concludes the proof of the observation. ◀