Expectation in Stochastic Games with Prefix-Independent Objectives

Laurent Doyen
Université Paris-Saclay, CNRS, ENS Paris-Saclay, LMF, 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 Mathematics of computing → Stochastic processes; Theory of computation → Probabilistic computation

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

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2025.16

Related Version *Full Version*: https://arxiv.org/abs/2405.18048 [21]

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 [19] 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:

Satisfaction. Given a threshold λ, to maximise the probability that φ-value of the outcome is greater than λ;

Expectation. To maximise the φ-value of the outcome in expectation.

Either view may be desirable depending on the context [5–8]. 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 [9]. Prefix-independent objectives also have the benefit that they satisfy the Bellman equations [34], 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 [35], 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 [16] 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 nondeterministic-

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

Objective	Complexity	Memory (Player 1) (lower [22], upper)	Memory (Player 2) (lower [22], upper)
FWMP(ℓ)	UP ∩ coUP	$\ell - 1$, ℓ	$\lvert V\rvert - \ell$, $\lvert V\rvert \cdot \ell$
BWMP	UP ∩ coUP	memoryless, memoryless	infinite, infinite

ally guessed, and we present a characterisation (Theorem 7, similar to [16, 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 [12]. As our secondary contribution, we illustrate our technique by applying it to two variants of window mean-payoff objectives: fixed (FWMP(ℓ)) and bounded (BWMP) window mean-payoff. Using our reduction, we are able to show that for both of these objectives, the expectation problem is in UP ∩ coUP (Theorem 18 and Theorem 22), a result that was not known before. The UP ∩ coUP upper bound for window mean-payoff objectives matches the special case of simple stochastic games [13, 20], 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 [12, 22]. 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 [38] where these games were studied under expectation semantics for discounted-sum objectives. In [14], 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, 30], and thus, are all in UP ∩ coUP [13]. A sub-exponential (or even quasi-polynomial) time deterministic algorithms for simple stochastic games on graphs with poly-logarithmic treewidth was proposed in [18]. In [29], sufficient conditions on the objective were shown such that optimal deterministic memoryless strategies exist for the players. In [34], 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 [24, 39], and with stochasticity in [28]. Finitary versions were introduced as window mean-payoff objectives [12]. For finitary mean-payoff objectives, the satisfaction problem [6] and the expectation problem [5] 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 [4, 36]. Both the expectation problem [5] and the satisfaction problem [6] for the FWMP(ℓ) objective are in PTIME, while they are in UP ∩ coUP for the BWMP objective. Ensuring the satisfaction and expectation semantics simultaneously was studied in MDPs for the mean-payoff objective in [17] and for the window mean-payoff objectives in [26]. 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 [22]. While positive and almost-sure satisfaction of $\mathsf{FWMP}(\ell)$ are in PTIME, it follows from [22] 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 [6].

Due to lack of space, some of the proofs and details have been omitted. A full version of the paper can be found in [21].

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 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 non-empty, 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 $\pi = v_0v_1v_2\cdots \in V^\omega$ 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* $v_iv_{i+1}\cdots v_j$ of π. Its length is $|\pi(i,j)| = j - i$, the number of edges. We denote by $\pi(0,j)$ the finite *prefix* $v_0v_1\cdots v_j$ of π, and by $\pi(i,\infty)$ the infinite *suffix* $v_iv_{i+1}\ldots$ 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

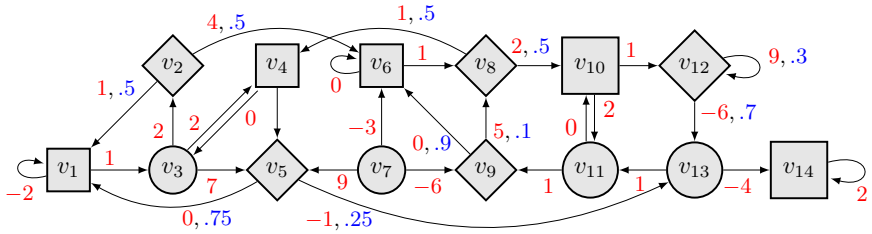

Figure 1 A stochastic game. Player 1 vertices are denoted by circles, Player 2 vertices are denoted by boxes, and probabilistic vertices are denoted by diamonds. The payoff for each edge is shown in red and probability distribution out of probabilistic vertices is shown in blue.

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 $\mathsf{Plays}_\mathcal{G}$ [35]. 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 [21].

Quantitative objectives. A *quantitative objective* is a Borel-measurable function of the form $\varphi \colon \mathsf{Plays}_\mathcal{G} \to \mathbb{R}$. 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 [33]. 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.

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 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 $\mathbb{E}^{\sigma}_v(\varphi)$ 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 $\mathsf{Pr}^{\sigma_1,\sigma_2}_{\mathcal{G},v}(\varphi)$. We only consider objectives φ that are Borel-measurable and whose image is bounded. Thus, by determinacy of Blackwell games [35], we have that stochastic games with objective φ are determined. That is, we have $\sup_{\sigma_1} \inf_{\sigma_2} \mathbb{E}^{\sigma}_v(\varphi) = \inf_{\sigma_2} \sup_{\sigma_1} \mathbb{E}^{\sigma}_v(\varphi)$. 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 the 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 $v \in V$.*

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

[1] We only consider the satisfaction and expectation of Borel-measurable objectives, and deterministic strategies suffice for such objectives [10]. 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.

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 [25] 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 [16, 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 [16, 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 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 $\vec{\mathsf{r}}$.

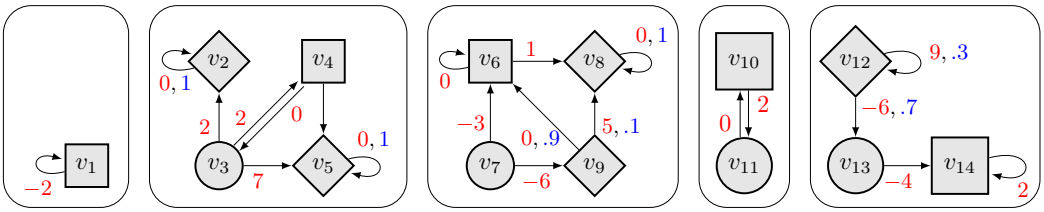

Figure 2 Restrictions $\mathcal{G}_{\mathsf{R}(1)}$, $\mathcal{G}_{\mathsf{R}(2)}$, $\mathcal{G}_{\mathsf{R}(3)}$, $\mathcal{G}_{\mathsf{R}(4)}$, and $\mathcal{G}_{\mathsf{R}(5)}$ of the game shown in Figure 1 for the vector $\vec{\mathsf{r}} = (-2, -1, -1, -1, -1, 0, 0, 0, 0, 1, 1, 2, 2, 2)$.

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 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 $1 \leq i \leq k_{\vec{\mathsf{r}}}$, 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 $1 \leq i_1, i_2 \leq k_{\vec{\mathsf{r}}}$ 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 $\vec{r}$ is a value vector that satisfies the **Bellman** condition, then for all $1 \leq i \leq k_{\vec{r}}$, 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 $\vec{r}$-class from which it does not exit. This follows from the Borel-Cantelli lemma [23] due to the fact that there is a positive probability to reach an $\vec{r}$-class without boundary vertices following a finite number of edges out of boundary vertices.

▶ **Proposition 5.** *Let $\vec{r}$ 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 $\vec{r}$-class as v. Then, with probability 1, the token eventually reaches a class $\mathsf{R}(j)$ for some $1 \leq j \leq k_{\vec{r}}$ 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 [25].

▶ **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 $\vec{r}$ to be equal to the expected φ-value vector $\vec{s}$. In addition to **Bellman**, Theorem 7 makes use of two more conditions, which we define before stating the theorem.

- ***lower-bound*** condition: for all $1 \leq i \leq k_{\vec{r}}$, 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 $1 \leq i \leq k_{\vec{r}}$, 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 $\vec{r}$, whose every component has denominator at most den_φ, that satisfies **Bellman**, **lower-bound**, and **upper-bound** is the expected φ-value vector $\vec{s}$.*

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

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 $\vec{s}$ satisfies the three conditions in Theorem 7.*

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

Suppose for the sake of contradiction that **lower-bound** does not hold, that is, there exists $1 \leq i \leq k_{\vec{s}}$ and a vertex v in $T^1_{\mathsf{S}(i)}$ such that Player 2 has a positive winning strategy from v for the $\{\varphi \leq \mathfrak{s}_i - \varepsilon_\varphi\}$ objective in $T^1_{\mathsf{S}(i)}$. Since $\{\varphi \leq \mathfrak{s}_i - \varepsilon_\varphi\}$ is a prefix-independent objective, from [9, Theorem 1] 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 $\{\varphi \leq \mathfrak{s}_i - \varepsilon_\varphi\}$ 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 $\{\varphi \leq \mathfrak{s}_i - \varepsilon_\varphi\}$.
- 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 $1 \leq i \leq k_{\vec{\mathsf{r}}}$, 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 $\vec{\mathsf{r}}$-class $\mathsf{R}(i)$ as v (i.e. a token only exits an $\vec{\mathsf{r}}$-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 $\vec{\mathsf{r}}$-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 $\vec{\mathsf{r}}$-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 $1 \leq j \leq k_{\vec{\mathsf{r}}}$, 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 $\vec{\mathsf{r}}$ satisfies **Bellman**, we have that $\sum_j p_j \mathfrak{r}_j = \mathfrak{r}_i$. Since $\vec{\mathsf{r}}$ 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 $\vec{\mathsf{s}} > \vec{\mathsf{r}} - \varepsilon_\varphi$. ◀

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 [20], 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 $1 \leq i \leq k_{\vec{\mathsf{r}}}$ 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 [16], 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 $q \leq (\widehat{\mathbb{P}})^{4\cdot|E|}$ 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 $\vec{\mathsf{s}}$-class without boundary vertices. If the token is in $\mathsf{S}(i)$ at some point in the play, then since $\vec{\mathsf{s}}$ 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 $\vec{\mathsf{s}}$-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 $\vec{\mathsf{s}}$-class.
- On the other hand, suppose $\mathsf{S}(i)$ is an $\vec{\mathsf{s}}$-class containing at least one boundary vertex, and let v be a boundary vertex in $\mathsf{S}(i)$. Then, since $\vec{\mathsf{s}}$ 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 $\vec{\mathsf{s}}$-class with boundary vertices can be expressed solely in terms of transition probabilities of the outgoing edges from boundary vertices and values of $\vec{\mathsf{s}}$-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 $\vec{\mathsf{s}}$-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 $\vec{\mathsf{s}}$-classes in the game by k instead of $k_{\vec{\mathsf{s}}}$ for the rest of this section. If every $\vec{\mathsf{s}}$-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 $\vec{\mathsf{s}}$-classes with boundary vertices, and therefore, there are $k - m$ $\vec{\mathsf{s}}$-classes without boundary vertices. Since there always exists at least one $\vec{\mathsf{s}}$-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 $\vec{\mathsf{s}}$-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 $\vec{\mathsf{s}}$-values of classes with boundary vertices, i.e., $\mathfrak{s}_1, \ldots, \mathfrak{s}_m$ in terms of $\vec{\mathsf{s}}$-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 $\vec{\mathsf{s}}$ 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 [32], 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 [12]:

(i) *fixed window mean-payoff* (FWMP(ℓ)) 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 [12] 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* $\varphi_{FWMP(\ell)}$*-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 [5, 6, 12], 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 [21] that simple stochastic games [20], which are in $\mathsf{UP} \cap \mathsf{coUP}$ [13], 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 $\mathsf{S}(i)$. We use the fact that the $\varphi_{\mathsf{FWMP}(\ell)}$-value of every play π is the largest λ such that, eventually, every λ-window in π closes in at most ℓ steps, and that λ is the mean payoff of a sequence of at most ℓ edges in π. To complete the argument, we show that if both players play optimally, then, with probability 1, the $\varphi_{\mathsf{FWMP}(\ell)}$-value of the outcome π is equal to $\mathfrak{s}_i$ and thus, $\mathfrak{s}_i$ is also of this form.

▶ **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 $\vec{\mathsf{s}}$-classes without boundary vertices. To see this, let $\widehat{w} = \max\{q \mid \exists p, q \in \mathbb{Z}, \exists e \in E : w(e) = \frac{p}{q} \text{ with } p, q \text{ co-prime}\}$ 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 $\vec{\mathsf{s}}$-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 $-W \cdot q \leq p \leq W \cdot q$.*

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 [22].

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 [22]. ◀

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 [22]. 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 following corollary of Lemma 19 states the bound $\underline{\mathsf{den}}_{\mathsf{BWMP}}$ for the φ_{BWMP} objective.

▶ **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 $-W \cdot q \leq p \leq W \cdot q$.*

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 [22], which in turn reduces to total supremum payoff [12], which is in $\mathsf{UP} \cap \mathsf{coUP}$ [27]. 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 $\mathsf{BWMP}(\lambda)$ [22]. Player 2 requires infinite memory in general for the $\mathsf{BWMP}(\lambda)$ objective even in non-stochastic games [12], 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 [16], which deals with the satisfaction of Boolean prefix-independent objectives. We also discuss practical implementations for window mean-payoff objectives and applicability of our technique to other prefix-independent objectives.

Comparison with [16]. In [16], 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 [16].

Applicability to other prefix-independent objectives. Recall that in order to be able to apply our characterisation to a prefix-independent objective φ, we require bounds W_φ on the image of φ and den_φ on the denominators of the optimal expected φ-values of vertices in the game. These bounds can easily be derived individually for the following quantitative prefix-independent objectives, and thus, our technique applies to these objectives.

Mean-payoff objective. The mean-payoff value of any play is bounded between the minimum and maximum edge payoffs in the game, directly giving bounds on the image of the mean-payoff objective. Since deterministic memoryless optimal strategies exist for both players [24], the expected mean-payoff value is a solution of a stationary distribution in the Markov chain obtained by fixing memoryless strategies of both players. This gives denominator bounds for the expected mean-payoff values of vertices in the game.

Limsup and liminf objectives. Since the liminf objective is equivalent to $\mathsf{FWMP}(\ell)$ objective with window length $\ell = 1$ [21], and the limsup objective is the dual of the liminf objective, our analysis with window mean-payoff objectives generalises limsup and liminf objectives.

Positive frequency payoff objective [29]. Here, each vertex has a payoff, and this objective returns the maximum payoff among all vertices that are visited with positive frequency in an infinite play. This objective is prefix-independent, as the frequency of a vertex is independent of finite prefixes. The image of the objective is bounded between the minimum and maximum vertex payoffs. We observe that the value of vertices in a value class without boundary vertices is equal to the payoff of a vertex in the class, (giving a denominator bound for these vertices), and Theorem 11 uses this to give a denominator bound for vertices in value classes with boundary vertices.

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 [15], which has been seen to be an anytime algorithm for the standard mean-payoff objective [34]. 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 [11, 15]. We describe the reduction in [21], 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 [34], 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 [37], optimistic value iteration [31], and topological value iteration [3] 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. 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.
4. C. Baier and J-P. Katoen. *Principles of model checking*. MIT Press, 2008.
5. 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.
6. 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, December 2020. doi:10.23638/LMCS-16(4:13)2020.
7. 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.
8. T. Brázdil, V. Brožek, K. Chatterjee, V. Forejt, and A. Kučera. Markov Decision Processes with Multiple Long-run Average Objectives. *Logical Methods in Computer Science*, Volume 10, Issue 1, Feb 2014. doi:10.2168/LMCS-10(1:13)2014.
9. K. Chatterjee. Concurrent games with tail objectives. *Theoretical Computer Science*, 388(1):181–198, 2007. doi:10.1016/j.tcs.2007.07.047.
10. 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.
11. K. Chatterjee, L. Doyen, and T. A. Henzinger. A Survey of Stochastic Games with Limsup and Liminf Objectives. In *ICALP*, pages 1–15. Springer, 2009. doi:10.1007/978-3-642-02930-1_1.
12. 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.
13. 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.

14 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.

15 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.

16 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.

17 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, July 2017. doi:10.23638/LMCS-13(2:15)2017.

18 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.

19 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.

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

21 L. Doyen, P. Gaba, and S. Guha. Expectation in Stochastic Games with Prefix-independent Objectives, 2024. doi:10.48550/arXiv.2405.18048.

22 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.

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

24 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.

25 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*. Online, 2024. doi:10.48550/arXiv.2305.10546.

26 P. Gaba and S. Guha. Optimising expectation with guarantees for window mean payoff in Markov decision processes. In *AAMAS*, pages 820–828. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 2025. doi:10.5555/3709347.3743600.

27 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.

28 D. Gillette. *Stochastic games with zero stop probabilities*, pages 179–188. Princeton University Press, December 1958.

29 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.

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

31 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.

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

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

34 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.

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

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

37 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.

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

39 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.