Stochastic Window Mean-Payoff Games

Laurent Doyen[1], Pranshu Gaba[2](✉), and Shibashis Guha[2]

[1] CNRS & LMF, ENS Paris-Saclay, Gif-sur-Yvette, France
ldoyen@lmf.cnrs.fr

[2] Tata Institute of Fundamental Research, Mumbai, India
{pranshu.gaba,shibashis}@tifr.res.in

Abstract. Stochastic two-player games model systems with an environment that is both adversarial and stochastic. The environment is modeled by a player (Player 2) who tries to prevent the system (Player 1) from achieving its objective. We consider finitary versions of the traditional mean-payoff objective, replacing the long-run average of the payoffs by payoff average computed over a finite sliding window. Two variants have been considered: in one variant, the maximum window length is fixed and given, while in the other, it is not fixed but is required to be bounded. For both variants, we present complexity bounds and algorithmic solutions for computing strategies for Player 1 to ensure that the objective is satisfied with positive probability, with probability 1, or with probability at least p, regardless of the strategy of Player 2. The solution crucially relies on a reduction to the special case of non-stochastic two-player games. We give a general characterization of prefix-independent objectives for which this reduction holds. The memory requirement for both players in stochastic games is also the same as in non-stochastic games by our reduction. Moreover, for non-stochastic games, we improve upon the upper bound for the memory requirement of Player 1 and upon the lower bound for the memory requirement of Player 2.

Keywords: Stochastic games · Finitary objectives · Mean-payoff · Reactive synthesis

1 Introduction

We consider two-player turn-based stochastic games played on graphs. Games are a central model in computer science, in particular for the verification and synthesis of reactive systems [18, 11, 17]. A stochastic game is played by two players on a graph with stochastic transitions, where the players represent the system and the adversarial environment, while the objective represents the functional requirement that the synthesized system aims to satisfy with a probability p as large as possible. The vertices of the graph are partitioned into system, environment, and probabilistic vertices. A stochastic game is played in infinitely many rounds, starting by initially placing a token on some vertex. In every round, if the token is on a system or an environment vertex, then the owner of the vertex chooses a successor vertex; if the token is on a probabilistic vertex, then the successor vertex is chosen according to a given probability distribution. The token moves to the successor vertex, from where the next round starts. The outcome is an infinite sequence of vertices, which is winning for the system if it satisfies the given objective. The associated *quantitative satisfaction problem* is to decide, given a threshold p, whether the system can win with probability at least p. The *almost-sure problem* is the special case where $p = 1$, and the *positive problem* is to decide whether the system can win with positive probability. The almost-sure and the positive problems are referred to as the *qualitative satisfaction problems.* When the answer to these decision problems is Yes, it is useful to construct a winning strategy for the system that can be used as a model for an implementation that ensures the objective be satisfied with the given probability.

Traditional objectives in stochastic games are ω-regular such as reachability, safety, and parity objectives [11], or quantitative such as mean-payoff objectives [16, 27]. For example, a parity objective may specify that every request of the environment is eventually granted by the system, and a mean-payoff objective may specify the long-run average power consumption of the system. A well-known drawback of parity and mean-payoff objectives is that only the long-run behaviour of the system may be specified [1, 9, 21], allowing weird transient behaviour: for example, a system may grant all its requests but with an unbounded response time; or a system with long-run average power consumption below some threshold may exhibit arbitrarily long (but finite) sequences with average power consumption above the threshold. This limitation has led to considering finitary versions of those objectives [9, 23, 8], where the sequences of undesired transient behaviours must be of fixed or bounded length. Window mean-payoff objectives [8] are quantitative finitary objectives that strengthen the traditional mean-payoff objective: the satisfaction of a window mean-payoff objective implies the satisfaction of the standard mean-payoff objective. Given a length $\ell \geq 1$, the fixed window mean-payoff objective ($\mathsf{FWMP}(\ell)$) is satisfied if except for a finite prefix, from every point in the play, there exists a window of length at most ℓ starting from that point such that the mean payoff of the window is at least a given threshold. In the bounded window mean-payoff objective (BWMP), it is sufficient that there exists some length ℓ for which the $\mathsf{FWMP}(\ell)$ objective is satisfied.

Contributions. We present algorithmic solutions for stochastic games with window mean-payoff objectives, and show that the positive and almost-sure problems are solvable in polynomial time for $\mathsf{FWMP}(\ell)$ (Theorem 5), and are in $\mathsf{NP} \cap \mathsf{coNP}$ for BWMP (Theorem 6). The complexity result for the almost-sure problem entails that the quantitative satisfaction problem is in $\mathsf{NP}\cap\mathsf{coNP}$ (for both the fixed and bounded version), using standard techniques for solving stochastic games with prefix-independent objectives [13]. Note that the $\mathsf{NP} \cap \mathsf{coNP}$ bound for the quantitative satisfaction problem matches the special case of reachability objectives in simple stochastic games [14], and thus would require a major breakthrough to be improved.

As a consequence, using the $\mathsf{FWMP}(\ell)$ objective instead of the standard mean-payoff objective provides a stronger guarantee on the system, and even with a polynomial complexity for the positive and the almost-sure problems (which is not known for mean-payoff objectives), and at no additional computational cost for the quantitative satisfaction problem. The solution relies on a reduction to non-stochastic two-player games (stochastic games without probabilistic vertices). The key result is to show that in order to win positively from some vertex of the game graph, it is necessary to win from some vertex of the non-stochastic game obtained by transforming all probabilistic vertices into adversarial vertices. While this condition, that we call the sure-almost-sure (SAS) property (Definition 1), was used to solve finitary Streett objectives [13], we follow a similar approach and generalize it to arbitrary prefix-independent objectives (Theorem 4). The bounds on the memory requirement of optimal strategies of Player 1 can also be derived from the key result, and are the same as optimal bounds for non-stochastic games. For the $\mathsf{FWMP}(\ell)$ and BWMP objectives in particular, we show that the memory requirement of Player 2 is also no more than the optimal memory required by winning strategies in non-stochastic games.

As solving a stochastic game with a prefix-independent objective φ reduces to solving non-stochastic games with objective φ and showing that φ satisfies the SAS property, we show that the $\mathsf{FWMP}(\ell)$ and BWMP objectives satisfy the SAS property (Lemma 4, Lemma 5) and rely on the solution of non-stochastic games with these objectives [8] to complete the reduction.

We improve the memory bounds for optimal strategies of both players in non-stochastic games. It is stated in [8] that $|V| \cdot \ell$ memory suffices for both players, where $|V|$ and ℓ are the number of vertices and the window length respectively. In [6, Theorem 2] and [19, Theorem 6.4], the bound is loosened to $\mathcal{O}(w_{\max} \cdot \ell^2)$ and $\mathcal{O}(w_{\max} \cdot \ell^2 \cdot |V|)$ for Player 1 and Player 2 respectively, where $w_{\max}$ is the maximum absolute payoff in the graph, as the original tighter bounds [8] were stated without proof. Since the payoffs are given in binary, this is exponential in the size of the input. In contrast, we tighten the bounds stated in [8]. We show that for Player 1, memory ℓ suffices (Theorem 1), and improve the bound on memory of Player 2 strategies as follows. We compute the set W of vertices from which Player 2 can ensure that the mean payoff remains negative for ℓ steps, as well as the vertices from which Player 2 can ensure that the game reaches W. These vertices are identified recursively on successive subgames of the original input game. If k is the number of recursive calls, then we show that $k \cdot \ell$ memory suffices for Player 2 to play optimally (Theorem 2). Note that $k \leq |V|$. We also provide a lower bound on the memory size for Player 2. Given $\ell \geq 2$, for every $k \geq 1$, we construct a graph with a set V of vertices such that Player 2 requires at least $k + 1 = \frac{1}{2}(|V| - \ell + 3)$ memory to play optimally (Theorem 3). This is an improvement over the result in [8] which showed that memoryless strategies do not suffice for Player 2. Our result is quite counterintuitive since given an open window (a window in which every prefix has a total payoff less than 0) that needs to be kept open for another $j \leq \ell$ steps from a vertex v, one would conjecture that it is sufficient for a Player 2 winning strategy to choose an edge from v that leads to the minimum payoff over paths of length j. Thus for every j, Player 2 should choose a fixed edge and hence memory of size ℓ should suffice. However, we show that this is not the case.

To the best of our knowledge, this work leads to the first study of stochastic games with finitary quantitative objectives.

Related work. Window mean-payoff objectives were first introduced in [8] for non-stochastic games, where solving $\mathsf{FWMP}(\ell)$ was shown to be in PTIME and BWMP in $\mathsf{NP} \cap \mathsf{coNP}$. These have also been studied for Markov decision processes (MDPs) in [4, 3]. In [4], a threshold probability problem has been studied, while in [3], the authors studied the problem of maximising the expected value of the window mean-payoff. Positive, almost-sure, and quantitative satisfaction of BWMP in MDPs are in $\mathsf{NP} \cap \mathsf{coNP}$ [4], the same as in non-stochastic games.

Parity objectives can be viewed as a special case of mean-payoff objectives [22]. A bounded window parity objective has been studied in [9, 20, 12] where a play satisfies the objective if from some point on, there exists a bound ℓ such that from every state with an odd priority, a smaller even priority occurs within at most ℓ steps. Non-stochastic games with bounded window parity objectives can be solved in PTIME [20, 12]. Stochastic games with bounded window parity objectives have been studied in [13] where the positive and almost-sure problems are in PTIME while the quantitative satisfaction problem is in $\mathsf{NP} \cap \mathsf{coNP}$. A fixed version of the window parity objective has been studied for two-player games and shown to be PSPACE-complete [26]. Another window parity objective has been studied in [5] for which both the fixed and the bounded variants have been shown to be in PTIME for non-stochastic games. The threshold problem for this objective has also been studied in the context of MDPs, and both fixed and bounded variants are in PTIME [4]. Finally, synthesis for *bounded* eventuality properties in LTL is 2-$\mathsf{EXPTIME}$-complete [23].

Due to lack of space, some of the proofs have been omitted. A full version of the paper can be found in [15].

2 Preliminaries

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 and Player 2. A *stochastic game* is a weighted directed graph $\mathcal{G} = ((V, E), (V_1, V_2, V_\Diamond), \mathbb{P}, w)$, 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 (in figures, Player 1 vertices are shown as circles, Player 2 vertices as boxes, and probabilistic vertices as diamonds, and we use pronouns "she/her" for Player 1 and "he/him" for Player 2);

- $\mathbb{P}\colon V_\Diamond \to \mathcal{D}(V)$, where $\mathcal{D}(V)$ is the set of probability distributions over V, is a transition function that maps probabilistic vertices $v \in V_\Diamond$ 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 nonzero probability); for the algorithmic and complexity results, we assume that probabilities are given as rational numbers.
- $w\colon E \to \mathbb{Q}$ 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 prefixes in $\mathcal{G}$ respectively; the symbol $\mathcal{G}$ is omitted when it can easily be derived from the context. We denote by $\mathsf{First}(\rho)$ and $\mathsf{Last}(\rho)$ the first vertex and the last vertex of a prefix $\rho \in \mathsf{Prefs}_\mathcal{G}$ respectively. We denote by $\mathsf{Prefs}^i_\mathcal{G}$ ($i \in \{1, 2\}$) the set of all prefixes ρ such that $\mathsf{Last}(\rho) \in V_i$.

Objectives. An *objective* φ is a Borel-measurable subset of $\mathsf{Plays}_\mathcal{G}$ [2]. A play $\pi \in \mathsf{Plays}_\mathcal{G}$ *satisfies* an objective φ if $\pi \in \varphi$. In a (zero-sum) stochastic game $\mathcal{G}$ with objective φ, the objective of Player 1 is φ, and the objective of Player 2 is the complement set $\overline{\varphi} = \mathsf{Plays}_\mathcal{G} \setminus \varphi$. Common examples of objectives are qualitative objectives such as reachability, safety, Büchi, and coBüchi.

An objective φ is *closed under suffixes* if for all plays π satisfying φ, all suffixes of π also satisfy φ, that is, $\pi(j, \infty) \in \varphi$ for all $j \geq 0$. An objective φ is *closed under prefixes* if for all plays π satisfying φ, for all prefixes ρ such that the concatenation $\rho \cdot \pi$ is a play in $\mathcal{G}$, i.e., $\rho \cdot \pi \in \mathsf{Plays}_\mathcal{G}$, we have that $\rho \cdot \pi \in \varphi$. An objective φ is *prefix-independent* if it is closed under both prefixes and suffixes. An objective φ is closed under suffixes if and only if the complement objective $\overline{\varphi}$ is closed under prefixes. Thus, an objective φ is prefix-independent if and only if its complement $\overline{\varphi}$ is prefix-independent.

Strategies. A (deterministic) *strategy* 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. The set of all strategies of Player $i \in \{1, 2\}$ in the game $\mathcal{G}$ is denoted by Λ_i. Strategies can be realised as the output of a (possibly infinite-state) Mealy machine. A *Mealy machine* is a deterministic transition system with transitions labelled by an input/output pair. 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. We point the reader to [15] for a description of how a strategy is defined by a Mealy machine.

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 $\rho, \rho' \in \mathsf{Prefs}_{\mathcal{G}}^{i}$ if $\mathsf{Last}(\rho) = \mathsf{Last}(\rho')$, then $\sigma_i(\rho) = \sigma_i(\rho')$. Memoryless strategies can be defined by Mealy machines with only one state.

A play $\pi = v_0 v_1 \cdots$ is *consistent* with a strategy $\sigma_i \in \Lambda_i$ ($i \in \{1,2\}$) if $v_{j+1} = \sigma_i(\pi(0,j))$ for all $j \geq 0$ such that $v_j \in V_i$. A play π is an *outcome* of σ_1 and σ_2 if it is consistent with both σ_1 and σ_2. We denote by $\mathsf{Pr}_{\mathcal{G},v}^{\sigma_1,\sigma_2}(\varphi)$ the probability that an outcome of σ_1 and σ_2 in $\mathcal{G}$ with initial vertex v satisfies φ.

Non-stochastic two-player games. A stochastic game without probabilistic vertices (that is, with $V_\Diamond = \varnothing$) is called a *non-stochastic two-player game* (or simply, non-stochastic game in the sequel). In a non-stochastic game $\mathcal{G}$ with objective φ, a strategy σ_i is *winning* from a vertex $v \in V$ for Player i ($i \in \{1,2\}$) if every play in $\mathcal{G}$ with initial vertex v that is consistent with σ_i satisfies the objective φ. A vertex $v \in V$ is *winning* for Player i in $\mathcal{G}$ if Player i has a winning strategy in $\mathcal{G}$ from v. The set of vertices in V that are winning for Player i in $\mathcal{G}$ is the *winning region* of Player i in $\mathcal{G}$, denoted $\langle\!\langle i \rangle\!\rangle_{\mathcal{G}}(\varphi)$. If a vertex v belongs to the winning region of Player i ($i \in \{1,2\}$), then Player i is said to play *optimally* from v if she follows a winning strategy.

Subgames. 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 out-neighbour in V', that is $E(v') \cap V' \neq \varnothing$, and ($ii$) every probabilistic vertex $v' \in V_\Diamond \cap V'$ has all out-neighbours 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'). We denote this subgame by $\mathcal{G} \restriction V'$. 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 $\mathsf{Plays}_{\mathcal{G}'} \cap \varphi$.

Satisfaction probability. A strategy σ_1 of Player 1 is *winning* with probability p from an initial vertex v in $\mathcal{G}$ for objective φ if $\mathsf{Pr}_{\mathcal{G},v}^{\sigma_1,\sigma_2}(\varphi) \geq p$ 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 φ if $\mathsf{Pr}_{\mathcal{G},v}^{\sigma_1,\sigma_2}(\varphi) > 0$ (resp., $\mathsf{Pr}_{\mathcal{G},v}^{\sigma_1,\sigma_2}(\varphi) = 1$) for all strategies σ_2 of Player 2. We refer to positive and almost-sure winning as *qualitative* satisfaction of φ, while for arbitrary $p \in [0,1]$, we call it *quantitative* satisfaction. We denote by $\langle\!\langle 1 \rangle\!\rangle_{\mathcal{G}}^{\mathsf{Pos}}(\varphi)$ (resp., by $\langle\!\langle 1 \rangle\!\rangle_{\mathcal{G}}^{\mathsf{AS}}(\varphi)$) the positive (resp., almost-sure) winning region of Player 1, i.e., the set of all vertices in $\mathcal{G}$ from which Player 1 has a positive (resp., almost-sure) winning strategy for $\mathcal{G}$ with objective φ. If a vertex v belongs to the positive (resp., almost-sure) winning region of 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.

Positive attractors and traps. The Player i *positive attractor* ($i \in \{1, 2\}$) to $T \subseteq V$, denoted $\mathsf{PosAttr}_i(T)$, is the set of vertices in V from which Player i can ensure that the token reaches a vertex in T with positive probability. It is possible to compute the positive attractor in $\mathcal{O}(|E|)$ time [10]. In non-stochastic games, a positive attractor to a set T is the same as an attractor to the set T, which we denote by $\mathsf{Attr}_i(T)$. Computation of $\mathsf{PosAttr}_i(T)$ gives a memoryless strategy for Player i that ensures that the token reaches T with positive probability. We call such a strategy a *positive-attractor strategy* of Player i.

A *trap* for Player 1 is a set $T \subseteq V$ such that for every vertex $v \in T$, if $v \in V_1 \cup V_\Diamond$, then $E(v) \subseteq T$, and if $v \in V_2$, then $E(v) \cap T \neq \varnothing$. In other words, from every vertex $v \in T$, Player 2 can ensure (with probability 1) that the token never leaves T, moreover using a memoryless strategy. A trap for Player 2 can be defined analogously.

Remark 1. Let $\mathcal{G}$ be a non-stochastic game with objective φ for Player 1. If φ is closed under suffixes, then the winning region of Player 1 is a trap for Player 2. As a corollary, if φ is prefix-independent, then the winning region of Player 1 is a trap for Player 2 and the winning region of Player 2 is a trap for Player 1.

3 Window mean payoff

We consider two types of window mean-payoff objectives, introduced in [8]: (*i*) *fixed window mean-payoff* objective ($\mathsf{FWMP}(\ell)$) in which a window length $\ell \geq 1$ is given, and (*ii*) *bounded window mean-payoff* objective (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 *total payoff* of an infix $\pi(i, i+n) = v_i v_{i+1} \cdots v_{i+n}$ is defined as $\mathsf{TP}(\pi(i, i+n)) = \sum_{k=i}^{i+n-1} w(v_k, v_{k+1})$. The *mean payoff* of an infix $\pi(i, i+n)$ is defined as $\mathsf{MP}(\pi(i, i+n)) = \frac{1}{n}\mathsf{TP}(\pi(i, i+n))$. Observe that the mean payoff of an infix is nonnegative if and only if the total payoff of the infix is nonnegative. The mean payoff of a play π is defined as $\mathsf{MP}(\pi) = \liminf\limits_{n\to\infty} \mathsf{MP}(\pi(0, n))$. Given a window length $\ell \geq 1$, a play $\pi = v_0 v_1 \cdots$ in $\mathcal{G}$ satisfies the *fixed window mean-payoff objective* $\mathsf{FWMP}_\mathcal{G}(\ell)$ if from every position after some point, it is possible to start an infix of length at most ℓ with a nonnegative mean payoff. Formally,

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

We omit the subscript $\mathcal{G}$ when it is clear from the context. Note that when $\ell = 1$, the $\mathsf{FWMP}(1)$ and $\overline{\mathsf{FWMP}(1)}$ (i.e., the complement of $\mathsf{FWMP}(1)$) objectives reduce to coBüchi and Büchi objectives respectively. The following properties of $\mathsf{FWMP}(\ell)$ have been observed in [8]. For all window lengths $\ell \geq 1$, if a play

π satisfies $\mathsf{FWMP}(\ell)$, then $\mathsf{MP}(\pi) \geq 0$. In all plays satisfying $\mathsf{FWMP}(\ell)$, there exists a suffix that can be decomposed into infixes of length at most ℓ, each with a nonnegative mean payoff. Such a desirable robust property is not guaranteed by the classical mean-payoff objective, where infixes of unbounded lengths may have negative mean payoff.

As defined in [8], given a play $\pi = v_0 v_1 \cdots$ and $0 \leq i < j$, we say that the window $\pi(i,j)$ is *open* if the total-payoff of $\pi(i,k)$ is negative for all $i < k \leq j$. Otherwise, the window is *closed*. Given $j > 0$, we say a window is open at j if there exists an open window $\pi(i,j)$ for some $i < j$. The window starting at position i *closes* at position j if j is the first position after i such that the total-payoff of $\pi(i,j)$ is nonnegative. If the window starting at i closes at j, then for all $i \leq k < j$, the windows $\pi(k,j)$ are closed. This property is called the *inductive property of windows*.

We also have the bounded window mean-payoff objective BWMP. A play π satisfies the BWMP objective if there exists a window length $\ell \geq 1$ for which π satisfies $\mathsf{FWMP}(\ell)$, i.e.,

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

Equivalently, a play π does not satisfy BWMP if for every suffix of π, for all $\ell \geq 1$, the suffix contains an open window of length ℓ. Note that both $\mathsf{FWMP}(\ell)$ for all $\ell \geq 1$ and BWMP are prefix-independent objectives.

Decision problems. Given a game $\mathcal{G}$, an initial vertex $v \in V$, a rational threshold $p \in [0,1]$, and an objective φ (that is either $\mathsf{FWMP}(\ell)$ for a given window length $\ell \geq 1$, or BWMP), consider the problem of deciding:

- *Positive satisfaction of* φ: whether Player 1 positively wins φ from v, i.e., whether $v \in \langle\!\langle 1 \rangle\!\rangle_{\mathcal{G}}^{\mathsf{Pos}}(\varphi)$.
- *Almost-sure satisfaction of* φ: whether Player 1 almost-surely wins φ from v, i.e., whether $v \in \langle\!\langle 1 \rangle\!\rangle_{\mathcal{G}}^{\mathsf{AS}}(\varphi)$.
- *Quantitative satisfaction of* φ (also known as *quantitative value problem* [13]): whether Player 1 wins φ from v with probability at least p, i.e., whether $\sup_{\sigma_1 \in \Lambda_1} \inf_{\sigma_2 \in \Lambda_2} \mathsf{Pr}_{\mathcal{G},v}^{\sigma_1,\sigma_2}(\varphi) \geq p$.

Note that these three problems coincide for non-stochastic games. As considered in previous works [8, 3, 4], the window length ℓ is usually small (typically $\ell \leq |V|$), and therefore we assume that ℓ is given in unary (while the payoff on the edges is given in binary). From determinacy of Blackwell games [24], stochastic games with window mean-payoff objectives as defined above are determined, i.e., the largest probability with which Player 1 is winning and the largest probability with which Player 2 is winning add up to 1.

Algorithms for non-stochastic window mean-payoff games. To compute the positive and almost-sure winning regions for Player 1 for $\mathsf{FWMP}(\ell)$, we recall intermediate objectives defined in [8]. The *good window* objective $\mathsf{GW}_{\mathcal{G}}(\ell)$ consists

Algorithm 1 $\mathsf{NonStocFWMP}(\mathcal{G}, \ell)$

```
In: $\mathcal{G} = ((V, E), (V_1, V_2, \varnothing), w)$ and $\ell \geq 1$
Out: $\langle\langle 1 \rangle\rangle_{\mathcal{G}}(\mathsf{FWMP}(\ell))$
1: $W_d \leftarrow \mathsf{NonStocDirFWMP}(\mathcal{G}, \ell)$
2: if $W_d = \varnothing$ then
3:     return $\varnothing$
4: else
5:     $A \leftarrow \mathsf{Attr}_1(W_d)$
6:     return $A \cup \mathsf{NonStocFWMP}(\mathcal{G} \restriction (V \setminus A), \ell)$
```

Algorithm 2 $\mathsf{NonStocDirFWMP}(\mathcal{G}, \ell)$

```
In: $\mathcal{G} = ((V, E), (V_1, V_2, \varnothing), w)$ and $\ell \geq 1$
Out: $\langle\langle 1 \rangle\rangle_{\mathcal{G}}(\mathsf{DirFWMP}(\ell))$
1: $W_{gw} \leftarrow \mathsf{GoodWin}(\mathcal{G}, \ell)$
2: if $W_{gw} = V$ or $W_{gw} = \varnothing$ then
3:     return $W_{gw}$
4: else
5:     $A \leftarrow \mathsf{Attr}_2(V \setminus W_{gw})$
6:     return $\mathsf{NonStocDirFWMP}(\mathcal{G} \restriction (W_{gw} \setminus A), \ell)$.
```

of all plays π in $\mathcal{G}$ such that the window opened at the first position in the play closes in at most ℓ steps:

$$\mathsf{GW}_{\mathcal{G}}(\ell) = \{\pi \in \mathsf{Plays}_{\mathcal{G}} \mid \exists j \in \{1, \ldots, \ell\} : \mathsf{MP}(\pi(0, j)) \geq 0\}$$

The *direct fixed window mean-payoff* objective $\mathsf{DirFWMP}_{\mathcal{G}}(\ell)$ consists of all plays π in $\mathcal{G}$ such that from every position in π, the window closes in at most ℓ steps:

$$\mathsf{DirFWMP}_{\mathcal{G}}(\ell) = \{\pi \in \mathsf{Plays}_{\mathcal{G}} \mid \forall i \geq 0 : \pi(i, \infty) \in \mathsf{GW}_{\mathcal{G}}(\ell)\}$$

The $\mathsf{FWMP}_{\mathcal{G}}(\ell)$ objective can be expressed in terms of $\mathsf{DirFWMP}_{\mathcal{G}}(\ell)$:

$$\mathsf{FWMP}_{\mathcal{G}}(\ell) = \{\pi \in \mathsf{Plays}_{\mathcal{G}} \mid \exists k \geq 0 : \pi(k, \infty) \in \mathsf{DirFWMP}_{\mathcal{G}}(\ell)\}$$

We refer to Algorithms 1, 2, and 3 from [8] shown here with the same numbering. They compute the winning regions for Player 1 for the $\mathsf{FWMP}(\ell)$, $\mathsf{DirFWMP}(\ell)$, and $\mathsf{GW}(\ell)$ objectives in non-stochastic games respectively. The original algorithms in [8] contain subtle errors for which the fixes are known [6, 19]. For completeness, we refer the reader to [15] for counterexamples for the algorithms in [8] along with brief explanations of correctness for the modified versions.

Algorithm 3 uses dynamic programming to compute, for all $v \in V$ and all lengths $i \in \{1, \ldots, \ell\}$, the largest payoff $C_i(v)$ that Player 1 can ensure from v within at most i steps. The winning region for $\mathsf{GW}(\ell)$ for Player 1 consists of all vertices v such that $C_\ell(v) \geq 0$.

4 Memory requirement for non-stochastic window mean-payoff games

The memory requirement for winning strategies of Player 1 in non-stochastic games with objective $\mathsf{FWMP}(\ell)$ is claimed to be $\mathcal{O}(|V| \cdot \ell)$ without proof [8, Lemma 7], and further "correctly stated" as $\mathcal{O}(w_{\max} \cdot \ell^2)$, where $w_{\max}$ is the maximum absolute payoff in the graph [6, Theorem 2]. We improve upon these bounds and show that memory of size ℓ suffices for a winning strategy of Player 1.

Algorithm 3 $\mathsf{GoodWin}(\mathcal{G}, \ell)$

```
In: $\mathcal{G} = ((V, E), (V_1, V_2, \varnothing), w)$ the non-stochastic game, and $\ell \geq 1$, the window length
Out: The set of vertices from which Player 1 wins $\mathsf{GW}(\ell)$ in $\mathcal{G}$
 1: for all $v \in V$ do
 2:     $C_0(v) \leftarrow 0$
 3:     for all $i \in \{1, \dots, \ell\}$ do
 4:         $C_i(v) \leftarrow -\infty$
 5: for all $i \in \{1, \dots, \ell\}$ do
 6:     for all $v \in V_1$ do
 7:         $C_i(v) \leftarrow \max_{(v,v') \in E}\{\max\{w(v, v'), w(v, v') + C_{i-1}(v')\}\}$
 8:     for all $v \in V_2$ do
 9:         $C_i(v) \leftarrow \min_{(v,v') \in E}\{\max\{w(v, v'), w(v, v') + C_{i-1}(v')\}\}$
10: $W_{gw} \leftarrow \{v \in V \mid C_\ell(v) \geq 0\}$
11: return $W_{gw}$
```

We also present a family of games with arbitrarily many vertices where Player 2 is winning and all his winning strategies require at least $\frac{1}{2}(|V| - \ell) + 3$ memory, while it was only known that memoryless strategies are not sufficient for Player 2 [8].

4.1 Memory requirement for Player 1 for FWMP objective

Upper bound on memory requirement for Player 1. We show that memory of size ℓ suffices for winning strategies of Player 1 for the $\mathsf{DirFWMP}(\ell)$ objective (Lemma 1), which in turn shows that the same memory also works for the $\mathsf{FWMP}(\ell)$ objective (Theorem 1).

Lemma 1. *If Player* 1 *wins in a non-stochastic game with objective* $\mathsf{DirFWMP}(\ell)$*, then Player* 1 *has a winning strategy with memory of size* ℓ*.*

Proof (Sketch). Given a non-stochastic game $\mathcal{G}$, let W_d be the winning region of Player 1 in $\mathcal{G}$ for objective $\mathsf{DirFWMP}(\ell)$. By definition, every vertex in W_d is also winning for Player 1 for the $\mathsf{GW}(\ell)$ objective.

A winning strategy σ_{d} of Player 1 in W_d satisfies the objective $\mathsf{GW}(\ell)$ by closing a window within at most ℓ steps and then restarts with the same strategy, playing for $\mathsf{GW}(\ell)$ and so on. Using memory space $Q = \{1, \dots, \ell\}$, we may store the number of steps remaining before the window must close. However, the window may close any time within ℓ steps, and the difficulty lies in detecting this independently of the history. For memory state $q = i$ and the next visited vertex being v, intuitively, the memory should be updated to $q = i - 1$ if the window did not close yet upon reaching v, and to $q = \ell$ if it did, but that depends on which path was followed to reach v (not just on v), which is not stored in the memory space.

The crux is to show that it is not always necessary for Player 1 to be able to infer when the window closes. Given the current memory state $q = i$, and the next visited vertex v, the memory update is as follows: if $C_i(v) \geq 0$ (that is, Player 1 can ensure the window from v will close within i steps), then we update to $q = i - 1$ (*decrement*) although the window may or may not have closed upon reaching v; otherwise $C_i(v) < 0$ and we update to $q = \ell - 1$ (*reset to ℓ and decrement*) and we show that in this case the window did close. Intuitively, updating to $q = i - 1$ is safe even if the window did close, because the strategy of Player 1 will anyway ensure the (upcoming) window is closed within $i - 1 < \ell$ steps. A formal description of a Mealy machine with ℓ states defining a winning strategy of Player 1 for the $\mathsf{DirFWMP}(\ell)$ objective is given in [15]. $\square$

Theorem 1. *If Player 1 wins in a non-stochastic game $\mathcal{G}$ with objective $\mathsf{FWMP}(\ell)$, then Player 1 has a winning strategy with memory of size ℓ.*

Proof (Sketch). Since $\mathsf{FWMP}(\ell)$ is a prefix-independent objective, we have that the winning region $\langle\langle 1 \rangle\rangle_{\mathcal{G}}(\mathsf{FWMP}(\ell))$ of Player 1 is a trap for Player 2 (Remark 1), and induces a subgame, say $\mathcal{G}_0$. Let there be $k + 1$ calls to the subroutine $\mathsf{NonStocDirFWMP}$ from Algorithm 1 where $k < |V|$. We denote by $(W_i)_{i \in \{1,\dots,k\}}$ the nonempty W_d returned by the i^{th} call to the subroutine, and let $A_i = \mathsf{Attr}_1(W_i)$. The A_i's are pairwise disjoint, and their union is $\bigcup_{i=1}^{k} A_i = \langle\langle 1 \rangle\rangle_{\mathcal{G}}(\mathsf{FWMP}(\ell))$. For $i \in \{1, \dots, k\}$, inductively define $\mathcal{G}_i$ to be the subgame induced by the complement of A_i in $\mathcal{G}_{i-1}$. Since $\mathsf{DirFWMP}(\ell)$ is closed under suffixes, for all $i \in \{1, \dots, k\}$, we have that W_i is a trap for Player 2 in $\mathcal{G}_i$ (Remark 1).

We construct a strategy σ_1^{NS} that follows the (memoryless) attractor strategy in $\bigcup_i (A_i \setminus W_i)$, and follows the winning strategy σ_{d} for $\mathsf{DirFWMP}(\ell)$ objective (defined in the proof of Lemma 1) in $\bigcup_i W_i$. The reader is pointed to [15] for a formal description of a Mealy machine defining the strategy σ_1^{NS}. For the correctness of the construction, the crux is to show that one of the sets W_i ($i \in \{1, \dots, k\}$) is never left from some point on. Intuitively, given the token is in A_i for some $i \in \{1, \dots, k\}$ (thus in $\mathcal{G}_i$), following σ_1^{NS}, the token will either remain in A_i, or leave the subgame $\mathcal{G}_i$ and enter A_j for a smaller index $j < i$. The result follows since this can be done at most k times. $\square$

Lower bound on memory requirement for Player 1. In [8], the authors show a game with $\ell = 4$ where Player 1 requires memory at least 3. This can be generalized to arbitrary ℓ to show that memory of size $\ell - 1$ may be necessary (See [15] for details).

4.2 Memory requirement for Player 2 for FWMP objective

Upper bound on memory requirement for Player 2. Now we show that for the $\overline{\mathsf{FWMP}(\ell)}$ objective, Player 2 has a winning strategy that uses memory of size at most $|V| \cdot \ell$. This has been loosely stated in [8] without a formal proof.

Theorem 2. *Let $\mathcal{G}$ be a non-stochastic game with objective $\overline{\mathsf{FWMP}(\ell)}$ for Player 2. Then, Player 2 has a winning strategy with memory size at most $|V| \cdot \ell$.*

Proof (Sketch). Since $\mathsf{FWMP}(\ell)$ is a prefix-independent objective, so is $\overline{\mathsf{FWMP}(\ell)}$. We have that $\langle\langle 2 \rangle\rangle_{\mathcal{G}}(\overline{\mathsf{FWMP}(\ell)})$ is a trap for Player 1 (Remark 1) and induces a subgame, say $\mathcal{H}_0$, of $\mathcal{G}$. Let there be $k+1$ calls to the subroutine $\mathsf{GoodWin}$ from Algorithm 2 (where $k < |V|$), and let $\mathcal{H}_i$ be the subgame corresponding to the i^{th} call of the subroutine. We denote by $(W_i)_{i \in \{1,\ldots,k\}}$ the complement of W_{gw} in $\mathcal{H}_i$, where W_{gw} is returned by the i^{th} call to the subroutine, and let $A_i = \mathsf{Attr}_2(W_i)$. The A_i's are pairwise disjoint, and their union is $\bigcup_{i=1}^{k} A_i = \langle\langle 2 \rangle\rangle_{\mathcal{G}}(\overline{\mathsf{FWMP}(\ell)})$.

We describe a winning strategy for the $\overline{\mathsf{FWMP}(\ell)}$ objective with memory $k \cdot \ell$, which is at most $|V| \cdot \ell$. The strategy is always in either *attractor mode* or *window-open mode*. When the game begins, it is in attractor mode. If the strategy is in attractor mode and the token is on a vertex $v \in A_i \setminus W_i$ for some $i \in \{1,\ldots,k\}$, then the attractor strategy is to eventually reach W_i. If the token reaches W_i, then the strategy switches to window-open mode. Since all vertices in W_i are winning for Player 2 for the $\overline{\mathsf{GW}(\ell)}$ objective, he can keep the window open for ℓ more steps, provided that Player 1 does not move the token out of the subgame $\mathcal{H}_i$. If, at some point, Player 1 moves the token out of the subgame $\mathcal{H}_i$ to A_j for a smaller index $j < i$, then the strategy switches back to attractor mode, this time trying to reach W_j in the bigger subgame $\mathcal{H}_j$. Otherwise, if Player 2 keeps the window open for ℓ steps, then the strategy switches back to attractor mode until the token reaches a vertex in $\bigcup_{i=1}^{k} W_i$. This strategy can be defined by a Mealy machine M_2^{NS} with states $\{1,\ldots,k\} \times \{1,\ldots,\ell\}$, where the first component tracks the smallest subgame $\mathcal{H}_i$ in which the window started to remain open, and the second component indicates how many more steps the window needs to be kept open for. A formal description of M_2^{NS} can be found in [15]. □

Lower bound on memory requirement for Player 2. In [8], it was shown that memoryless strategies do not suffice for Player 2. We improve upon this lower bound. Given a window length $\ell \geq 2$, for every $k \geq 1$, we construct a game $\mathcal{G}_{k,\ell}$ with $2k + \ell - 1$ vertices such that every winning strategy of Player 2 in $\mathcal{G}_{k,\ell}$ requires at least $k+1$ memory.

Theorem 3. *There exists a family of non-stochastic games* $\{\mathcal{G}_{k,\ell}\}_{k \geq 1, \ell \geq 2}$ *with objective* $\mathsf{FWMP}(\ell)$ *for Player* 1 *and edge weights in* $\{-1, 0, +1\}$ *such that every winning strategy of Player* 2 *requires at least* $\frac{1}{2}(|V| - \ell + 1) + 1$ *memory, where* $|V| = 2k + \ell - 1$.

Proof (Sketch). Let $A = \{a_1,\ldots,a_k\}$, $B = \{b_1,\ldots,b_k\}$, and $C = \{c_1,\ldots,c_{\ell-1}\}$ be pairwise disjoint sets. The vertices of $\mathcal{G}_{k,\ell}$ are $A \cup B \cup C$ with $V_1 = A \cup C$ and $V_2 = B$. Figure 1 shows the game $\mathcal{G}_{4,3}$. A more formal description of $\mathcal{G}_{k,\ell}$ can be found in [15].

Observe that the only open windows of length ℓ in the game $\mathcal{G}_{k,\ell}$ are sequences of the form $a_p b_r c_{\ell-1} \cdots c_1$ for all $p \leq r$. Also note that Player 2 has a winning strategy that wins starting from every vertex in the game, as Player 2 can force the token to eventually take a red edge followed by two black edges.

When the token reaches a vertex $b_r \in B$, Player 2 can either move the token to $a_r \in A$ or to $c_{\ell-1} \in C$. Depending on which vertex the token was on before reaching b_r, one of the two choices is *good* for Player 2. If the token reaches b_r

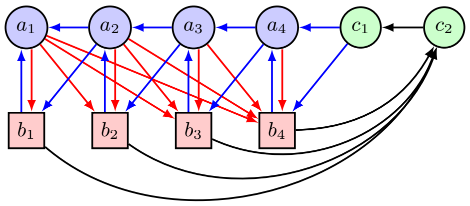

Figure 1: Game $\mathcal{G}_{4,3}$ with parameter $k = 4$ and window length $\ell = 3$. Red edges (from a_p to b_r for $p \le r$) have payoff -1, black edges (from b_r to c_2) have payoff 0, and blue edges (the remaining edges) have payoff $+1$.

Table 1: Good choices $\chi(u, b_r)$ for all $u \in A \cup \{c_1\}$ and $b_r \in B$ in the game $\mathcal{G}_{4,3}$

$a_1b_1 \to c_2$	$a_2b_1 \to a_1$			
$a_1b_2 \to c_2$	$a_2b_2 \to c_2$	$a_3b_2 \to a_2$		
$a_1b_3 \to c_2$	$a_2b_3 \to c_2$	$a_3b_3 \to c_2$	$a_4b_3 \to a_3$	
$a_1b_4 \to c_2$	$a_2b_4 \to c_2$	$a_3b_4 \to c_2$	$a_4b_4 \to c_2$	$c_1b_4 \to a_4$

from a_p for $p \le r$, then it is *good* for Player 2 to move the token to $c_{\ell-1} \in C$ so that the window starting at a_p remains open for ℓ steps. Otherwise, if the token reaches b_r from a_{r+1}, then it is *good* for Player 2 to move the token to a_r so that an edge with negative payoff may eventually be taken. For all $u \in A \cup \{c_1\}$, for all $b_r \in B$ such that (u, b_r) is an edge in $\mathcal{G}_{k,\ell}$, we denote by $\chi(u, b_r)$ the vertex a_r or $c_{\ell-1}$ that is good for Player 2. We list the good choices in the game $\mathcal{G}_{4,3}$ in Table 1. The columns are indexed by $u \in A \cup \{c_1\}$ and the rows are indexed by $b_r \in B$.

We show that for each column in the table, there exists a distinct memory state in every Mealy machine defining a winning strategy of Player 2. This gives a lower bound of $k + 1$ on the number of states of such a Mealy machine. Since $\mathcal{G}_{k,\ell}$ has $2k + \ell - 1$ vertices, the memory requirement of a winning strategy of Player 2 is at least $\frac{1}{2}(|V| - \ell + 1) + 1$. □

Given a winning strategy σ_2^{NS} of Player 2 for the $\overline{\mathsf{FWMP}(\ell)}$ objective, the following lemma gives an upper bound on the number of steps between consecutive open windows of length ℓ in any play consistent with σ_2^{NS}. This lemma is used in Section 6, where we construct an almost-sure winning strategy of Player 2 for the $\overline{\mathsf{FWMP}(\ell)}$ objective.

Lemma 2. *Let $\mathcal{G}$ be a non-stochastic game such that $\langle\langle 2 \rangle\rangle_{\mathcal{G}}(\overline{\mathsf{FWMP}(\ell)}) = V$. Let σ_2^{NS} be a finite-memory strategy of Player 2 of memory size M that is winning for $\overline{\mathsf{FWMP}(\ell)}$ from all vertices in $\mathcal{G}$. Then, for every play π of $\mathcal{G}$ consistent with σ_2^{NS}, every infix of π of length $\mathsf{M} \cdot |V| \cdot \ell$ contains an open window of length ℓ.*

The proof is based on the pigeonhole principle and appears in [15].

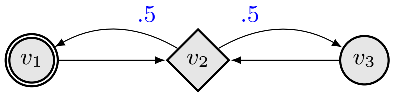

Figure 2: Büchi objective does not satisfy the SAS property in this game.

5 Reducing stochastic games to non-stochastic games

For a stochastic game $\mathcal{G}$, let $\mathcal{G}_{\mathsf{NS}} = ((V, E), (V_1, V_2 \cup V_\Diamond, \varnothing), w)$ be the *(adversarial) non-stochastic game corresponding to* $\mathcal{G}$, obtained by changing all probabilistic vertices in $\mathcal{G}$ to Player 2 vertices. In [13], a property of finitary Streett objective was used to solve stochastic games by reducing them to non-stochastic games with the same objective. In this section, we generalize this property for arbitrary prefix-independent objectives.

Definition 1 (Sure-almost-sure (SAS) property). *A prefix-independent objective φ in a game $\mathcal{G}$ satisfies the SAS property if $\langle\!\langle 2 \rangle\!\rangle_{\mathcal{G}_{\mathsf{NS}}}(\overline{\varphi}) = V$ implies $\langle\!\langle 2 \rangle\!\rangle^{\mathsf{AS}}_{\mathcal{G}}(\overline{\varphi}) = V$, that is, if Player 2 wins the objective $\overline{\varphi}$ from every vertex in $\mathcal{G}_{\mathsf{NS}}$, then Player 2 almost-surely wins the same objective $\overline{\varphi}$ from every vertex in $\mathcal{G}$.*

Every prefix-independent objective satisfies the converse of the SAS property since if Player 2 even wins positively from all vertices in $\mathcal{G}$, then since he controls all probabilistic vertices in $\mathcal{G}_{\mathsf{NS}}$, he wins from all vertices in $\mathcal{G}_{\mathsf{NS}}$ by choosing optimal successors of probabilistic vertices. We show in Section 6 that for all stochastic games $\mathcal{G}$, the objectives $\mathsf{FWMP}(\ell)$ and BWMP satisfy the SAS property, while in Example 1, we show that there exists a stochastic game in which Büchi objective does not satisfy the SAS property.

Example 1. Consider the game $\mathcal{G}$ in Figure 2. The objective φ in this game is a Büchi objective: a play π satisfies the Büchi objective if π visits vertex v_1 infinitely often. Although from every vertex, with positive probability (in fact, with probability 1), a play visits v_1 infinitely often, from none of the vertices, Player 1 can ensure the Büchi objective in the non-stochastic game $\mathcal{G}_{\mathsf{NS}}$.

Theorem 4 gives complexity bounds for solving stochastic games with objectives satisfying the SAS property in terms of the complexity of solving non-stochastic games with the same objective.

Theorem 4. *Given $\mathcal{G}$ and φ, suppose in every subgame $\mathcal{G}'$ of $\mathcal{G}$, the objective φ restricted to $\mathcal{G}'$ satisfies the SAS property. Let $\mathsf{NonStocWin}_\varphi(\mathcal{G}_{\mathsf{NS}})$ be an algorithm computing $\langle\!\langle 1 \rangle\!\rangle_{\mathcal{G}_{\mathsf{NS}}}(\varphi)$ in $\mathcal{G}_{\mathsf{NS}}$ in time $\mathbb{C}$. Then, the positive and almost-sure satisfaction of φ can be decided in time $\mathcal{O}(|V| \cdot (\mathbb{C} + |E|))$ and $\mathcal{O}(|V|^2 \cdot (\mathbb{C} + |E|))$ respectively.*

Moreover, for positive and almost-sure satisfaction of φ, the memory requirement for Player 1 to play optimally in stochastic games is no more than that for non-stochastic games.

Algorithm 4 $\mathsf{PosWin}_\varphi(\mathcal{G})$

```
In: $\mathcal{G} = ((V, E), (V_1, V_2, V_\Diamond), \mathbb{P}, w)$ and $\varphi$
Out: $\langle\langle 1 \rangle\rangle_{\mathcal{G}}^{\mathsf{Pos}}(\varphi)$
1: $W_1 \leftarrow \mathsf{NonStocWin}_\varphi(\mathcal{G}_{\mathsf{NS}})$
2: if $W_1 = \varnothing$ then
3:     return $\varnothing$
4: else
5:     $A_1 \leftarrow \mathsf{PosAttr}_1(W_1)$
6:     return $A_1 \cup \mathsf{PosWin}_\varphi(\mathcal{G} \upharpoonright (V \setminus A_1))$
```

Algorithm 5 $\mathsf{ASWin}_\varphi(\mathcal{G})$

```
In: $\mathcal{G} = ((V, E), (V_1, V_2, V_\Diamond), \mathbb{P}, w)$ and $\varphi$
Out: $\langle\langle 1 \rangle\rangle_{\mathcal{G}}^{\mathsf{AS}}(\varphi)$
1: $W_2 \leftarrow V \setminus \mathsf{PosWin}_\varphi(\mathcal{G})$
2: if $W_2 = \varnothing$ then
3:     return $V$
4: else
5:     $A_2 \leftarrow \mathsf{PosAttr}_2(W_2)$
6:     return $\mathsf{ASWin}_\varphi(\mathcal{G} \upharpoonright (V \setminus A_2))$
```

Theorem 4 does not give bounds on the memory requirement for winning strategies of Player 2 for objective φ in the stochastic game, but we provide such bounds specifically for $\mathsf{FWMP}(\ell)$ and BWMP in Section 6. We give a sketch of the proof of Theorem 4 below. The complete proof appears in [15].

The algorithms to compute the positive and almost-sure winning regions in $\mathcal{G}$, and their proofs of correctness are the same as in the case of finitary Streett objectives described in [13]. The PosWin_φ algorithm (Algorithm 4) uses $\mathsf{NonStocWin}_\varphi$ as a subroutine to compute $\langle\langle 1 \rangle\rangle_{\mathcal{G}}^{\mathsf{Pos}}(\varphi)$. The fact that φ satisfies the SAS property is used to show the correctness of this algorithm. The depth of recursive calls of this algorithm is bounded above by $|V|$, which gives the complexity bound. The ASWin_φ algorithm (Algorithm 5) in turn uses PosWin_φ as a subroutine to compute the $\langle\langle 1 \rangle\rangle_{\mathcal{G}}^{\mathsf{AS}}(\varphi)$. The depth of recursive calls of this algorithm is also bounded above by $|V|$, which gives the complexity bound. The following lemma, which is a special case of Theorem 1 in [7], is used to show the correctness of this algorithm.

Lemma 3. *[7, Theorem 1] For a stochastic game $\mathcal{G}$ with prefix-independent objective φ, if $\langle\langle 2 \rangle\rangle_{\mathcal{G}}^{\mathsf{Pos}}(\varphi) = V$, then $\langle\langle 2 \rangle\rangle_{\mathcal{G}}^{\mathsf{AS}}(\varphi) = V$.*

For both positive and almost-sure winning, Player 1 does not require any additional memory in the stochastic game compared to the non-stochastic game. We describe a strategy σ_1^{Pos} of Player 1 that is positive winning from all vertices in $\langle\langle 1 \rangle\rangle_{\mathcal{G}}^{\mathsf{Pos}}(\varphi)$. In each recursive call to PosWin_φ algorithm, from every vertex in W_1, the strategy σ_1^{Pos} mimics a winning strategy of Player 1 in $\mathcal{G}_{\mathsf{NS}}$, while for vertices in $A_1 \setminus W_1$, it follows a memoryless attractor strategy to reach W_1. The same strategy is almost-sure winning for Player 1 from all vertices in $\langle\langle 1 \rangle\rangle_{\mathcal{G}}^{\mathsf{AS}}(\varphi)$.

Finally, we look at the quantitative decision problem. The quantitative satisfaction for φ can be decided in NP^B ([13, Theorem 6]), where B is an oracle deciding positive and almost-sure satisfaction problems for φ. It is not difficult to see that the quantitative satisfaction for φ can be decided in $\mathsf{NP}^B \cap \mathsf{coNP}^B$. Moreover, from the proof of [13, Theorem 6], it follows that the memory requirement of winning strategies for both players for the quantitative decision problem is no greater than that for the qualitative decision problem.

Corollary 1. *Given $\mathcal{G}$ and φ as described in Theorem 4, let B be an oracle deciding the qualitative satisfaction of φ. Then, the quantitative satisfaction of φ is in $\mathsf{NP}^B \cap \mathsf{coNP}^B$. Moreover, the memory requirement of optimal strategies for both players is no greater than that for the positive and almost-sure satisfaction of φ.*

6 Reducing stochastic window mean-payoff games: A special case

In this section, we show that for all stochastic games $\mathcal{G}$ and for all $\ell \geq 1$, the objectives $\mathsf{FWMP}_{\mathcal{G}}(\ell)$ and $\mathsf{BWMP}_{\mathcal{G}}$, which are prefix-independent, satisfy the SAS property of Definition 1. Thus, by Theorem 4, we obtain bounds on the complexity and memory requirements of Player 1 for the positive and almost-sure satisfaction of these objectives. We also show that for both these objectives, the memory requirements of Player 2 to play optimally for positive and almost-sure winning in stochastic games is no more than that of the non-stochastic games. The algorithms to compute the positive and almost-sure winning regions of Player 1 for both $\mathsf{FWMP}(\ell)$ and BWMP objectives are obtained by instantiating φ equal to $\mathsf{FWMP}(\ell)$ and BWMP in Algorithms 4 and 5. Thus, we obtain the algorithms $\mathsf{PosWin}_{\mathsf{FWMP}(\ell)}$, $\mathsf{ASWin}_{\mathsf{FWMP}(\ell)}$, $\mathsf{PosWin}_{\mathsf{BWMP}}$, and $\mathsf{ASWin}_{\mathsf{BWMP}}$.

6.1 Fixed window mean-payoff objective

We first discuss the SAS property for the $\mathsf{FWMP}(\ell)$ objective.

Lemma 4. *In stochastic games, for all $\ell \geq 1$, the $\mathsf{FWMP}(\ell)$ objective satisfies the SAS property.*

Proof (Sketch). We show that for all stochastic games $\mathcal{G}$, if $\langle\!\langle 2 \rangle\!\rangle_{\mathcal{G}_{\mathsf{NS}}}(\overline{\mathsf{FWMP}(\ell)}) = V$, then $\langle\!\langle 2 \rangle\!\rangle_{\mathcal{G}}^{\mathsf{AS}}(\overline{\mathsf{FWMP}(\ell)}) = V$. If $\langle\!\langle 2 \rangle\!\rangle_{\mathcal{G}_{\mathsf{NS}}}(\overline{\mathsf{FWMP}(\ell)}) = V$, then from Theorem 2, there exists a finite-memory strategy σ_2^{NS} (say, with memory M) of Player 2 that is winning for objective $\overline{\mathsf{FWMP}(\ell)}$ from every vertex in $\mathcal{G}_{\mathsf{NS}}$. Given such a strategy, we construct below a strategy σ_2^{AS} of Player 2 in the stochastic game $\mathcal{G}$ that is almost-sure winning for $\overline{\mathsf{FWMP}(\ell)}$ from every vertex in $\mathcal{G}$.

In $\mathcal{G}_{\mathsf{NS}}$, Player 2 controls vertices in $V_2 \cup V_\Diamond$, while in $\mathcal{G}$, Player 2 only controls vertices in V_2 and the probability function $\mathbb{P}$ determines the successors of vertices in $V_\Diamond$. While the strategy σ_2^{NS} is winning for $\overline{\mathsf{FWMP}(\ell)}$ from all vertices in $\mathcal{G}_{\mathsf{NS}}$, it may not be almost-sure winning for $\overline{\mathsf{FWMP}(\ell)}$ in $\mathcal{G}$. This is because each time the token is on a probabilistic vertex, a *deviation* occurs with positive probability, i.e., the successor chosen by the distribution is not consistent with σ_2^{NS}, resulting in a potentially worse outcome for Player 2. For example, in Figure 3, we see a stochastic game $\mathcal{G}$ and a Mealy machine M_2^{NS} defining a strategy σ_2^{NS} that is winning for Player 2 from all vertices in the non-stochastic game $\mathcal{G}_{\mathsf{NS}}$. In all outcomes in $\mathcal{G}_{\mathsf{NS}}$ that are consistent with σ_2^{NS}, the token never moves from v_6 to v_7. However, in $\mathcal{G}$, a deviation may lead the token to move along (v_6, v_7). This

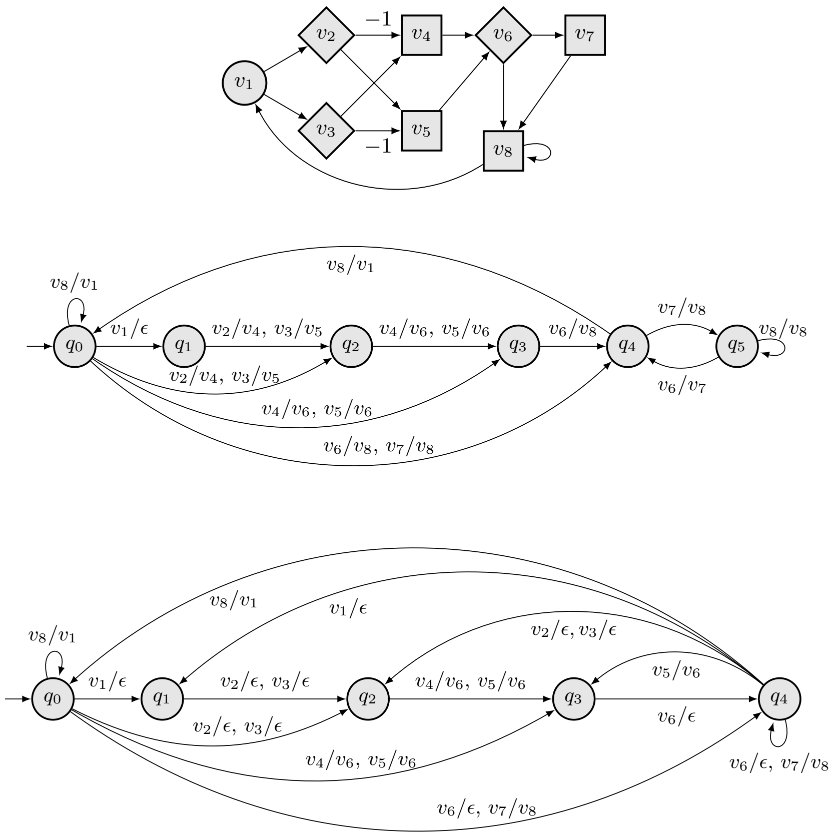

Figure 3: (top) Stochastic game $\mathcal{G}$ with objective $\overline{\mathsf{FWMP}(3)}$ for Player 2. All unlabelled edges have payoff 0. (middle) Mealy machine M_2^{NS} defining a strategy σ_2^{NS} that is winning from all vertices in $\mathcal{G}_{\mathsf{NS}}$ for $\overline{\mathsf{FWMP}(3)}$. (bottom) Part of the Mealy machine M_2^{AS} defining a reset strategy that is almost-sure winning from all vertices in $\mathcal{G}$.

results in a losing outcome for Player 2 as the token gets trapped in v_8, and subsequently no window remains open for ℓ steps. Such harmful deviations can be detected, and starting with the strategy σ_2^{NS}, we construct a strategy σ_2^{AS} that mimics σ_2^{NS} as long as harmful deviations do not occur, and *resets* otherwise, i.e., the strategy forgets the prefix of the play before the deviation. For instance, when the token moves from v_6 to v_7 in $\mathcal{G}$, the strategy resets and the play continues as if the game began from v_7. We call σ_2^{AS} a *reset strategy*. Figure 3 shows a part of a Mealy machine M_2^{AS} defining a reset strategy for the game $\mathcal{G}$. The figure contains all the reset transitions out of q_4, but the reset transitions out of q_1, q_2, and q_3 have been omitted for space. More details on how to obtain a Mealy machine that defines σ_2^{AS} from a Mealy machine that defines σ_2^{NS} without adding any new states can be found in [15].

Now, we argue that the reset strategy is almost-sure winning for Player 2 from all vertices in $\mathcal{G}$. If a play in $\mathcal{G}$ continues for $\mathsf{M} \cdot |V| \cdot \ell$ steps without deviating, then by Lemma 2, it contains an open window of length ℓ. From any point in the play, the probability that σ_2^{AS} successfully copies σ_2^{NS} for i steps (that is, no deviations occur) is at least p^i, where p is the minimum probability over all the edges in $\mathcal{G}$. It follows that from every point in the play, the probability that an open window of length ℓ occurs in the next $\mathsf{M} \cdot |V| \cdot \ell$ steps is at least $p^{\mathsf{M}\cdot|V|\cdot\ell}$. Therefore, from every position in the play, the probability that an open window of length ℓ occurs eventually is at least $\sum_{i\geq 0}(1-p^{\mathsf{M}\cdot|V|\cdot\ell})^i \cdot p^{\mathsf{M}\cdot|V|\cdot\ell} = 1$. Thus, with probability 1, infinitely many open windows of length ℓ occur in the outcome, and the outcome satisfies $\overline{\mathsf{FWMP}(\ell)}$. Thus, all vertices in $\mathcal{G}$ are almost-sure winning for Player 2 for $\overline{\mathsf{FWMP}(\ell)}$. For all stochastic games $\mathcal{G}$, the objective $\mathsf{FWMP}(\ell)$ satisfies the SAS property. $\square$

We now construct a strategy σ_2^{Pos} of Player 2 that is positive winning from all vertices in $\langle\!\langle 2 \rangle\!\rangle_{\mathcal{G}}^{\mathsf{Pos}}(\overline{\mathsf{FWMP}(\ell)})$. Let W_2^i and A_2^i denote the sets W_2 and A_2 computed in the i^{th} recursive call of the $\mathsf{ASWin}_{\mathsf{FWMP}(\ell)}$ algorithm respectively. If the token is in $\bigcup_i W_2^i$, then σ_2^{Pos} mimics σ_2^{AS}; if the token is in $\bigcup_i A_2^i \setminus W_2^i$, then σ_2^{Pos} is a positive-attractor strategy to W_2^i which is memoryless. Then, σ_2^{Pos} is a positive winning strategy for Player 2 from all vertices in $\langle\!\langle 2 \rangle\!\rangle_{\mathcal{G}}^{\mathsf{Pos}}(\overline{\mathsf{FWMP}(\ell)})$. Using Theorem 4, Corollary 1, and Lemma 4, we have the following.

Theorem 5. *Given a stochastic game $\mathcal{G}$, a window length $\ell \geq 1$, and a threshold $p \in [0,1]$, for $\mathsf{FWMP}_{\mathcal{G}}(\ell)$, the positive and almost-sure satisfaction for Player 1 are in PTIME, and the quantitative satisfaction is in $\mathsf{NP} \cap \mathsf{coNP}$. Moreover for optimal strategies, memory of size ℓ is sufficient for Player 1 and memory of size $|V| \cdot \ell$ is sufficient for Player 2.*

6.2 Bounded window mean-payoff objective

We show that the SAS property holds for the BWMP objective for all stochastic games $\mathcal{G}$.

Lemma 5. *In stochastic games, the BWMP objective satisfies the SAS property.*

Proof (Sketch). We show that for all stochastic games $\mathcal{G}$, if $\langle\!\langle 2 \rangle\!\rangle_{\mathcal{G}_{\mathsf{NS}}}(\overline{\mathsf{BWMP}}) = V$, then $\langle\!\langle 2 \rangle\!\rangle_{\mathcal{G}}^{\mathsf{AS}}(\overline{\mathsf{BWMP}}) = V$. Since every play that satisfies $\overline{\mathsf{BWMP}}$ also satisfies $\overline{\mathsf{FWMP}(\ell)}$ for all $\ell \geq 1$, if $\langle\!\langle 2 \rangle\!\rangle_{\mathcal{G}_{\mathsf{NS}}}(\overline{\mathsf{BWMP}}) = V$, then $\langle\!\langle 2 \rangle\!\rangle_{\mathcal{G}_{\mathsf{NS}}}(\overline{\mathsf{FWMP}(\ell)}) = V$. It follows that for each $\ell \geq 1$, Player 2 has a finite-memory strategy (say, with memory M_ℓ), that is winning for the $\overline{\mathsf{FWMP}(\ell)}$ objective from all vertices in $\mathcal{G}_{\mathsf{NS}}$. For every such strategy, we construct a reset strategy σ_2^ℓ of memory size at most M_ℓ as described in the proof of Lemma 4 that is almost-sure winning for the $\overline{\mathsf{FWMP}(\ell)}$ objective from all vertices. We use these strategies to construct an infinite-memory strategy σ_2^{AS} of Player 2 that is almost-sure winning for $\overline{\mathsf{BWMP}}$ from all vertices in the stochastic game $\mathcal{G}$.

Let p be the minimum probability over all edges in the game, and for all $\ell \geq 1$, let $q(\ell)$ denote $p^{\mathsf{M}_\ell\cdot|V|\cdot\ell}$. We partition a play of the game into phases $1, 2, \ldots$ such that for all $\ell \geq 1$, the length of phase ℓ is equal to $\mathsf{M}_\ell \cdot |V| \cdot \ell \cdot \lceil 1/q(\ell) \rceil$. We define the strategy σ_2^{AS} as follows: if the game is in phase ℓ, then σ_2^{AS} is σ_2^ℓ, the reset strategy that is almost-sure winning for $\overline{\mathsf{FWMP}(\ell)}$ in $\mathcal{G}$.

We show that σ_2^{AS} is almost-sure winning for Player 2 for $\overline{\mathsf{BWMP}}$ in $\mathcal{G}$. Let E_ℓ denote the event that phase ℓ contains an open window of length ℓ. Given a play π, if E_ℓ occurs in π for infinitely many $\ell \geq 1$, then for every suffix of π and for all $\ell \geq 1$, the suffix contains an open window of length ℓ, and π satisfies $\overline{\mathsf{BWMP}}$. For all $\ell \geq 1$, we compute the probability that E_ℓ occurs in the outcome. For all $\ell \geq 1$, we can divide phase ℓ into $\lceil 1/q(\ell) \rceil$ blocks of length $\mathsf{M}_\ell \cdot |V| \cdot \ell$ each. If at least one of these blocks contains an open window of length ℓ, then the event E_ℓ occurs. It follows from the proof of Lemma 4 that if Player 2 follows σ_2^ℓ, then the probability that there exists an open window of length ℓ in the next $\mathsf{M}_\ell \cdot |V| \cdot \ell$ steps is at least $q(\ell)$. Hence, the probability that none of the blocks in the phase contains an open window of length ℓ is at most $(1 - q(\ell))^{\lceil 1/q(\ell) \rceil}$. Thus, the probability that E_ℓ occurs in phase ℓ is at least $1 - (1 - q(\ell))^{\lceil 1/q(\ell) \rceil} > 1 - \frac{1}{e} \approx 0.63 > 0$. It follows that with probability 1, for infinitely many values of $\ell \geq 1$, the event E_ℓ occurs in π. ☐

Note that solving a non-stochastic game with the BWMP objective is in $\mathsf{NP} \cap \mathsf{coNP}$ [8]. Thus by Corollary 1, quantitative satisfaction for BWMP is in $\mathsf{NP}^{\mathsf{NP} \cap \mathsf{coNP}} \cap \mathsf{coNP}^{\mathsf{NP} \cap \mathsf{coNP}}$, which is the same as $\mathsf{NP} \cap \mathsf{coNP}$ [25].

Moreover, from [8], Player 1 has a memoryless strategy and Player 2 needs infinite memory to play optimally in non-stochastic games with the BWMP objective. From the proof of Lemma 5, by using the strategy σ_2^{AS}, Player 2 almost-surely wins $\overline{\mathsf{BWMP}}$ from all vertices in $\langle\!\langle 2 \rangle\!\rangle_{\mathcal{G}}^{\mathsf{AS}}(\overline{\mathsf{BWMP}})$. We can construct a positive winning strategy σ_2^{Pos} for Player 2 from all vertices in $\langle\!\langle 2 \rangle\!\rangle_{\mathcal{G}}^{\mathsf{Pos}}(\overline{\mathsf{BWMP}})$ in a similar manner as done for the positive winning strategy for $\overline{\mathsf{FWMP}(\ell)}$ in Section 6.1. We summarize the results in the following theorem:

Theorem 6. *Given a stochastic game $\mathcal{G}$ and a threshold $p \in [0,1]$, for $\mathsf{BWMP}_{\mathcal{G}}$, the positive, almost-sure, and quantitative satisfaction for Player 1 are in $\mathsf{NP} \cap \mathsf{coNP}$. Moreover, a memoryless strategy suffices for Player 1, while Player 2 requires an infinite memory strategy to play optimally.*

Acknowledgement. We thank Mickael Randour for pointing out reference [6], making us aware of the bugs in the algorithms of [8] and the correct version of these algorithms as presented here. This work is partially supported by the Indian Science and Engineering Research Board (SERB) grant SRG/2021/000466 and by the Indo-French Centre for the Promotion of Advanced Research (IFCPAR).

References

1. Alur, R., Henzinger, T.A.: Finitary fairness. In: LICS. pp. 52–61. IEEE Computer Society (1994)
2. Baier, C., Katoen, J.: Principles of model checking. MIT Press (2008)

3. Bordais, B., Guha, S., Raskin, J.F.: Expected window mean-payoff. In: FSTTCS. LIPIcs, vol. 150, pp. 32:1–32:15 (2019)
4. Brihaye, T., Delgrange, F., Oualhadj, Y., Randour, M.: Life is Random, Time is Not: Markov Decision Processes with Window Objectives. Logical Methods in Computer Science **Volume 16, Issue 4** (Dec 2020)
5. Bruyère, V., Hautem, Q., Randour, M.: Window parity games: an alternative approach toward parity games with time bounds. In: GandALF. EPTCS, vol. 226, pp. 135–148 (2016)
6. Bruyère, V., Hautem, Q., Raskin, J.F.: On the complexity of heterogeneous multidimensional games. In: CONCUR. LIPIcs, vol. 59, pp. 11:1–11:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016)
7. Chatterjee, K.: Concurrent games with tail objectives. Theoretical Computer Science **388**(1), 181–198 (2007)
8. Chatterjee, K., Doyen, L., Randour, M., Raskin, J.F.: Looking at mean-payoff and total-payoff through windows. Information and Computation **242**, 25–52 (2015)
9. Chatterjee, K., Henzinger, T.A.: Finitary winning in ω-regular games. In: TACAS. pp. 257–271. LNCS 3920, Springer (2006)
10. Chatterjee, K., Henzinger, T.A.: Value iteration. In: 25 Years of Model Checking - History, Achievements, Perspectives. pp. 107–138. LNCS 5000, Springer (2008)
11. Chatterjee, K., Henzinger, T.A.: A survey of stochastic ω-regular games. Journal of Computer and System Sciences **78**(2), 394–413 (2012)
12. Chatterjee, K., Henzinger, T.A., Horn, F.: Finitary winning in omega-regular games. ACM Trans. Comput. Log. **11**(1), 1:1–1:27 (2009)
13. Chatterjee, K., Henzinger, T.A., Horn, F.: Stochastic games with finitary objectives. In: MFCS. pp. 34–54. Springer Berlin Heidelberg (2009)
14. Condon, A.: The complexity of stochastic games. Information and Computation **96**(2), 203–224 (1992)
15. Doyen, L., Gaba, P., Guha, S.: Stochastic window mean-payoff games. CoRR **abs/2304.11563** (2023), https://arxiv.org/abs/2304.11563
16. Ehrenfeucht, A., Mycielski, J.: Positional strategies for mean payoff games. Int. Journal of Game Theory **8**(2), 109–113 (1979)
17. Filar, J., Vrieze, K.: Competitive Markov Decision Processes. Springer (1997)
18. Grädel, E., Thomas, W., Wilke, T. (eds.): Automata, Logics, and Infinite Games: A Guide to Current Research. LNCS 2500, Springer (2002)
19. Hautem, Q.: The Complexity of Combining Objectives in Two-Player Games. Ph.D. thesis, Université de Mons (2018)
20. Horn, F.: Faster algorithms for finitary games. In: TACAS. Lecture Notes in Computer Science, vol. 4424, pp. 472–484. Springer (2007)
21. Horn, F., Thomas, W., Wallmeier, N., Zimmermann, M.: Optimal strategy synthesis for request-response games. RAIRO Theor. Informatics Appl. **49**(3), 179–203 (2015)
22. Jurdzinski, M.: Deciding the winner in parity games is in UP $\cap$ co-UP. Inf. Process. Lett. **68**(3), 119–124 (1998)
23. Kupferman, O., Piterman, N., Vardi, M.Y.: From liveness to promptness. Formal Methods Syst. Des. **34**(2), 83–103 (2009)
24. Martin, D.A.: The determinacy of blackwell games. The Journal of Symbolic Logic **63**(4), 1565–1581 (1998), http://www.jstor.org/stable/2586667
25. Schöning, U.: A low and a high hierarchy within NP. Journal of Computer and System Sciences **27**(1), 14–28 (1983)
26. Weinert, A., Zimmermann, M.: Easy to win, hard to master: Optimal strategies in parity games with costs. In: CSL. LIPIcs, vol. 62, pp. 31:1–31:17 (2016)

27. Zwick, U., Paterson, M.: The complexity of mean payoff games on graphs. Theor. Comput. Sci. **158**(1&2), 343–359 (1996)