Optimising Expectation with Guarantees for Window Mean Payoff in Markov Decision Processes

Pranshu Gaba
Tata Institute of Fundamental Research
Mumbai, India
pranshu.gaba@tifr.res.in

Shibashis Guha
Tata Institute of Fundamental Research
Mumbai, India
shibashis.guha@tifr.res.in

ABSTRACT

The window mean-payoff objective strengthens the classical mean-payoff objective by computing the mean-payoff over a finite window that slides along an infinite path. 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. In this paper, we look at the problem of synthesising strategies in Markov decision processes that maximise the window mean-payoff value in expectation, while also simultaneously guaranteeing that the value is above a certain threshold. We solve the synthesis problem for three different kinds of guarantees: sure (that needs to be satisfied in the worst-case, that is, for an adversarial environment), almost-sure (that needs to be satisfied with probability one), and probabilistic (that needs to be satisfied with at least some given probability p).

We show that for the fixed window mean-payoff objective, when the window length is given in unary, all the three problems are in PTIME, while for the bounded window mean-payoff objective, they are in NP ∩ coNP, and thus have the same complexity as for maximising the expected performance without any guarantee. Moreover, we show that pure finite-memory strategies suffice for maximising the expectation with sure and almost-sure guarantees, whereas, for maximising expectation with a probabilistic guarantee, randomised strategies are necessary in general.

KEYWORDS

Beyond worst-case synthesis; reactive synthesis; finitary objectives; mean payoff; Markov decision processes; two-player games

ACM Reference Format:
Pranshu Gaba and Shibashis Guha. 2025. Optimising Expectation with Guarantees for Window Mean Payoff in Markov Decision Processes. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Beyond worst-case synthesis. Classical two-player quantitative zero-sum games [2, 23] involve decision making against a purely antagonistic environment, where a minimum performance needs to be guaranteed even in the worst case. On the other hand, Markov decision processes (MDPs) [25] model uncertainty, and decision making involves ensuring a higher expected performance against a stochastic environment which usually does not provide any guarantee on the worst-case performance. Both these models have their own weaknesses. A strategy against an adversarial environment may provide worst-case guarantee but may be suboptimal in its expected behaviour. On the other hand, a strategy that maximises the expected performance may fail miserably in the worst case.

However, in practice, both might be desired simultaneously: A system needs to provide guarantee in the worst-case, and perform well in an expected sense against a stochastic environment. In [10], the beyond worst-case (BWC) framework was introduced to provide strict worst-case guarantee as well as good expected performance. In particular, the study was made for two quantitative objectives: mean payoff and shortest path. While this work focussed on the restricted class of finite memory strategies, it was also shown that infinite memory strategies are strictly more powerful than finite-memory strategies in the BWC setting [10]. The synthesis of infinite memory strategies was subsequently studied in [18].

Window mean payoff. For Boolean and quantitative prefix-independent objectives specified as the limit of a reward function in the long run [20, 27], a play may satisfy such an objective and yet also exhibit undesired behaviours for arbitrarily long intervals [11, 14]. Finitary or window objectives strengthen such prefix-independent objectives by restricting the undesired behaviour to intervals of bounded length (windows) of the play. As a particular case, we consider the *window mean-payoff objectives*, which are finitary versions of the classical mean-payoff objective. Window mean-payoff objectives [11] are quantitative finitary objectives that strengthen the classical mean-payoff objective: the satisfaction of a window mean-payoff objective implies the satisfaction of the classical mean-payoff objective. Given a length $\ell \geq 1$, the fixed window mean-payoff objective ($\mathsf{FWMP}(\ell, \lambda)$) 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 λ. In the bounded window mean-payoff objective ($\mathsf{BWMP}(\lambda)$), it is sufficient that there exists some length ℓ for which the $\mathsf{FWMP}(\ell, \lambda)$ objective is satisfied. The value of an outcome run is the largest (supremum) γ such that from some point on, the run can be decomposed into windows of size smaller than the fixed or existentially quantified bound ℓ, all having a mean-payoff value at least γ.

Contributions. In this paper, we study three different problems to maximise expectation while simultaneously providing guarantees for the fixed and the bounded window mean-payoff objectives. Given an MDP and thresholds $\alpha, \beta \in \mathbb{Q}$, synthesise a strategy that:

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1. (Beyond worst case (BWC) synthesis) (i) ensures a window mean-payoff at least α surely, i.e. against all strategies of an adversarial environment, and (ii) an expectation that is at least β against a stochastic model of the environment.

2. (Beyond probability threshold (BPT) synthesis) (i) ensures a window mean-payoff at least α with at least some given probability p, and (ii) an expectation that is at least β against a stochastic model of the environment.

3. (Beyond almost sure (BAS) synthesis) (i) ensures a window mean-payoff at least α almost surely, i.e. with probability one, and (ii) an expectation that is at least β against a stochastic model of the environment.

Motivating examples. We consider some motivating examples in the context of window mean payoff for maximising expectation while providing guarantees at the same time. **1.** *Consistent output of power plant.* A power plant may be required to output 10 MW of power on average. A plant that outputs 240 MW of power for an hour and 0 MW for the rest of the day satisfies the requirement but is not desirable if there is no means to store the energy. Using windows, we can ask for an output of 10 MW every hour. Moreover, a power plant may have multiple ways to generate power (solar, wind, hydro) with different rates and reliability, e.g. solar output is high during the day and low during the night, while hydro is consistently low-moderate. It is may be desired to devise a strategy that maximises the output in expectation while producing sufficient power at all times for critical applications such as hospitals and trains. **2.** *Investment in stock market.* While investing in the stock market, an investor not only wants a higher expected return, but may as well prefer to be risk-averse, that is, the stocks do not crash or such events happen rarely. Further, the investor may consistently want to receive returns over a certain period or a window of time. **3.** *Gambling* [12]. In gambling, while the goal is to maximise the expected profit, a desirable policy may want to avoid risk and try to ensure that the chance of losing is less than a certain probability. Further, a gambler would like to receive a profit amount from time to time and the payment should not be deferred indefinitely.

We show that, for the case of fixed window mean-payoff, if the window length is given in unary, then all of the above problems are in PTIME (Theorems 4.3, 4.6, and 4.7) and are thus no more complex than solving two-player games [11] or maximising expectation for the same objective in an MDP [7]. For classical mean-payoff objective, we note that the first problem above is in NP ∩ coNP [6] while the second and the third problems are in PTIME [18]. For the case of bounded window mean-payoff objective, all the above problems are in NP ∩ coNP (Theorem 5.1), thus showing that the results are no more complex than solving two-player games or maximising expectation for the same objective in an MDP [7, 11].

Our techniques are different from the BWC synthesis for classical mean-payoff objective. While both our work as well as [10] reason on the so-called end-components (ECs), the approach for the BWC synthesis problem for classical mean payoff in [10] relies on a special kind of end-component called winning end-components (WECs). These are ECs where all vertices are winning for the sure mean-payoff objective while staying inside the end-component. In the current work, for window mean-payoff objective, for the BWC synthesis problem, we do not need to find the WECs but instead it suffices to compute the maximum sure window mean-payoff value from each vertex. Further, unlike mean-payoff objective [10], for window mean-payoff objective infinite memory strategies are no more powerful than finite memory strategies and we need to switch from a strategy corresponding to the expectation maximisation to the sure satisfaction strategy only once.

Related Work. The beyond worst-case framework was introduced in [10] for quantitative objectives. The problem was studied for finite-memory strategies and it was shown to be in NP ∩ coNP for mean-payoff objective. The case of infinite memory strategy for the BWC synthesis problem was left open in [10] and was solved in [18]. Further, in [18], a natural relaxation of the BWC problem, the beyond almost-sure synthesis problem (BAS) was introduced and was shown to be in PTIME for mean-payoff objective. The beyond probability threshold synthesis problem was studied for mean-payoff objective in [15]. In [6], the BWC, BAS, and BPT synthesis problems were studied for qualitative omega-regular objectives encoded as parity objectives. The problems were shown to be in NP ∩ coNP. In [17], the above problem was studied in the context of stochastic games which are a generalisation of MDPs where the environment is both stochastic and adversarial. A combination of optimising expected mean payoff and surely satisfying omega-regular [1], safety [22], and energy objectives [8] were also considered. In [4], Boolean combinations of objectives that are omega-regular properties that need to be enforced either surely, almost surely, existentially, or with non-zero probability were studied. It was shown that both randomisation and infinite memory may be required by an optimal strategy. In [5], a combination of parity objective and multiple reachability objectives along with threshold probabilities were considered where the parity objective needs to be satisfied surely and each reachability objective is satisfied with the corresponding threshold probability. The BWC and the BPT problems were also studied for the discounted-sum objective in partially observed MDPs (POMDPs) [12, 16].

Mean-payoff objectives were studied initially in two-player games, without stochasticity [20, 27], and finitary versions were introduced as window mean-payoff objectives [11]. For finitary mean-payoff objectives, the satisfaction problem [9] and the expectation problem [7] were studied in MDPs. Both the expectation problem [7] and the satisfaction problem [9] for the FWMP(ℓ) objective are in PTIME, while they are in UP ∩ coUP for the BWMP objective. The satisfaction problem for window mean-payoff objectives has been studied recently in [19] for stochastic games.

In the current work too, we analyse ECs, but in a way that is different from the above works. While pure finite-memory strategies suffice for the BWC and the BAS synthesis problems for the window mean-payoff objectives, we need finite-memory randomised strategies for the BPT synthesis problem.

A full version of the paper with complete proofs appear in [21].

2 TECHNICAL PRELIMINARIES

Probability distributions. For a finite set A, a *probability distribution* over A is a function $\Pr\colon A \to [0,1]$ such that $\sum_{a\in A} \Pr(a) = 1$. We denote by $\mathcal{D}(A)$ the set of all probability distributions over A. The support of the probability distribution Pr on A is $\mathrm{Supp}(\Pr) = \{a \in A \mid \Pr(a) > 0\}$. For algorithm and complexity reasons, we assume that the probability distributions take rational values.

Markov decision processes. A *Markov decision process* (MDP) is a tuple $\mathcal{M} = ((V, E), (V_\circ, V_\diamond), \mathbb{P}, w)$ where:

- (V, E) is called the *arena* of $\mathcal{M}$. It is a directed graph with a set V of vertices and a set $E \subseteq (V_\circ \times V_\diamond) \cup (V_\diamond \times V_\circ)$ of edges such that for each vertex $v \in V$, there is an out-edge from v in the game (i.e., no deadlocks). We denote by $E(v)$ the set of vertices u such that $(v, u) \in E$. We say that the MDP $\mathcal{M}$ is finite if the set V is finite. Unless mentioned otherwise, we consider MDPs to be finite in this work.
- $(V_\circ, V_\diamond)$ is a partition of the set V of vertices, where $V_\circ$ denotes the set of vertices belonging to the player and $V_\diamond$ denotes probabilistic vertices.
- $\mathbb{P}\colon V_\diamond \to \mathcal{D}(V_\circ)$ is the *probability function* that returns the probability distribution over the out-neighbours of probabilistic vertices. We require for every $v \in V_\diamond$ that $\mathsf{Supp}(\mathbb{P}(v)) = E(v)$, that is, for all vertices $v' \in V$, we have that $\mathbb{P}(v)(v') > 0$ if and only if v' is an out-neighbour of v.
- $w\colon E \to \mathbb{Z}$ is the *payoff function* that defines an integer payoff for every edge in the arena. Let $W_\mathcal{M}$ be the maximum weight appearing on the edges in $\mathcal{M}$. We drop the subscript when it is clear from the context.

With a little abuse of nomenclature, we mean by *self-loop from a vertex v* a sequence of two edges starting from and ending at v so that player vertices and probabilistic vertices alternate. A payoff λ on the self-loop here denotes that both the edges that are part of the self-loop have the same payoff λ.

A *run* of the MDP begins by placing a token on an initial vertex which is a player vertex and proceeds in steps. In each step, if the token is on a player vertex v, then the player chooses an out-edge of v and moves the token along that edge. Otherwise, if the token is on a probabilistic vertex v, then the out-edge is chosen by the probability distribution $\mathsf{Pr}(v)$. This continues *ad infinitum*, resulting in a run π that is an infinite path in the arena.

For a run $\pi = v_0 v_1 v_2 \cdots$, we denote by $\pi(i)$ the vertex v_i, by $\pi(i, j)$ the infix $v_i \cdots v_j$, by $\pi(0, j)$ the finite prefix $v_0 v_1 \cdots v_j$, and by $\pi(i, \infty)$ the infinite suffix $v_i v_{i+1} \cdots$. The length of an infix $\pi(i, j)$ is the number of edges, that is $j - i$. We denote by $\mathsf{Runs}^\mathcal{M}$ and $\mathsf{Prefs}_\circ^\mathcal{M}$ the set of all runs in $\mathcal{M}$ and the set of all finite prefixes in $\mathcal{M}$ ending in a vertex in $V_\circ$ respectively. We drop the superscript $\mathcal{M}$ when they are clear from the context. For a prefix $\rho \in \mathsf{Prefs}_\circ$, we denote by $\mathsf{Last}(\rho)$ the last vertex of ρ. We denote by $\inf(\pi)$ the set of vertices in V that occur infinitely often in π.

An MDP where every vertex in $V_\circ$ has exactly one out-neighbour is called a *Markov chain*. Figure 1 shows an example of an MDP. In figures, in MDPs, we denote player vertices by circles and probabilistic vertices by diamonds.

Boolean objectives. Depending on the specifications, some runs are desirable for the player, and some are not. A *Boolean objective* φ is a set of runs that are desirable for the player. We say a run $\pi \in \mathsf{Runs}$ *satisfies* an objective φ if $\pi \in \varphi$. Given a set $T \subseteq V$ of target vertices, a common Boolean objective is the *reachability objective*, defined as $\mathsf{Reach}(T) = \{\pi \in \mathsf{Runs} \mid \exists i \geq 0, \pi(i) \in T\}$, i.e., the set of runs that visit T.

Quantitative objectives. A *quantitative objective* is a function $\varphi\colon \mathsf{Runs} \to \mathbb{Q} \cup \{\pm\infty\}$ that assigns to each run in the MDP a numerical value that denotes how good the run is for the player. Some common examples of quantitative objectives include mean-payoff, discounted-sum payoff, energy payoff, total payoff and liminf payoff. For a run $\pi = v_0 v_1 v_2 \cdots$, the liminf mean-payoff objective is defined as follows: $\varphi_{\mathsf{MP}}(\pi) \coloneqq \liminf_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n} w(v_i, v_{i+1})$. The objectives studied in this paper, the window mean-payoff objectives (defined later in this section), are also quantitative objectives. Corresponding to a quantitative objective φ, we define *threshold Boolean objectives* $\{\pi \in \mathsf{Runs} \mid \varphi(\pi) \geq \lambda\}$, for thresholds $\lambda \in \mathbb{R}$. We denote these objectives succinctly as $\{\varphi \geq \lambda\}$.

A quantitative objective φ is *closed under suffixes* if for all runs π, and for all suffixes $\pi(j, \infty)$ of π, we have $\varphi(\pi) = \varphi(\pi(j, \infty))$. An objective φ is *closed under prefixes* if for all runs π and all prefixes ρ such that $\rho \cdot \pi \in \mathsf{Runs}^\mathcal{M}$, we have $\varphi(\pi) = \varphi(\rho \cdot \pi)$. An objective φ is *prefix-independent* if it is closed under both prefixes and suffixes. Mean-payoff is an example of a prefix-independent objective. Prefix independence can be defined analogously for Boolean objectives.

Strategies. A *strategy* for the player in an MDP $\mathcal{M}$ is a function $\sigma : \mathsf{Prefs}_\circ \to \mathcal{D}(V)$ that maps prefixes ending in a vertex $v \in V_\circ$ to a distribution over the successors of v. The set of all strategies of the player in the MDP $\mathcal{M}$ is denoted by $\Lambda_\mathcal{M}$. 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 input/output pairs. Intuitively, in each step, if the token is on a vertex v that belongs to the player, then v is given as input to the Mealy machine, and the Mealy machine outputs a distribution over the successor probabilistic vertices of v that the player must choose. Otherwise, the token is on a vertex v that is a probabilistic vertex, in which case, the Mealy machine outputs the distribution $\mathbb{P}(v)$ that is part of the MDP $\mathcal{M}$.

A strategy is *deterministic* if for every prefix $\rho \in \mathsf{Prefs}_\circ$, we have that $|\mathsf{Supp}(\sigma(\rho))| = 1$, otherwise it is *randomised*. The *memory size* of a strategy σ is the smallest number of states a Mealy machine defining σ can have. A strategy σ is *memoryless* if for all prefixes $\rho, \rho' \in \mathsf{Prefs}_\circ$, if $\mathsf{Last}(\rho) = \mathsf{Last}(\rho')$, then $\sigma(\rho) = \sigma(\rho')$. Memoryless strategies can be defined by Mealy machines with only one state. Fixing a strategy σ of the player in an MDP yields a (possibly infinite-state) Markov chain, and we represent this by $\mathcal{M}^\sigma$.

A run $\pi = v_0 v_1 \cdots$ is *consistent* with a strategy $\sigma \in \Lambda_\mathcal{M}$ if for all $j \geq 0$ with $v_j \in V$, we have $v_{j+1} \in \mathsf{Supp}(\sigma(\pi(0, j)))$. A run π is an *outcome* of a strategy σ if π is consistent with σ. We denote by $\mathsf{Out}_v^\mathcal{M}(\sigma)$ the set of runs of $\mathcal{M}$ that start from v and are consistent with strategy σ.

Satisfaction probability of Boolean objectives. For a Boolean objective φ, we denote by $\mathsf{Pr}^\sigma_{\mathcal{M},v}(\varphi)$ the probability that an outcome of the strategy σ in $\mathcal{M}$ with initial vertex v satisfies φ. This probability is defined for cones defined by finite prefixes, and it extends to infinite runs uniquely by Carathéodory's extension theorem [26].

Given an MDP $\mathcal{M}$ with a Boolean objective φ, starting from a vertex v in $\mathcal{M}$, we are interested in finding the maximum probability with which the player can ensure that the objective φ is satisfied. In the decision problem, we ask if given $p \in [0, 1] \cap \mathbb{Q}$, does there exist a strategy $\sigma \in \Lambda_\mathcal{M}$ of the player such that $\mathsf{Pr}^\sigma_{\mathcal{M},v}(\varphi) \geq p$.

Expected value of quantitative objectives. For a quantitative objective φ, we are interested in determining the maximum value of φ that the player can ensure in expectation. Formally, given a strategy σ and an initial vertex v, we denote by $\mathbb{E}^\sigma_{\mathcal{M},v}(\varphi)$

the *expected φ-value of an outcome* of σ from v, that is, the expectation of φ over all plays with initial vertex v under the probability measure $\Pr^{\sigma}_{\mathcal{M},v}(\varphi)$. The expected φ-value of a vertex v is $\mathbb{E}_{\mathcal{M},v}(\varphi) = \sup_{\sigma\in\Lambda_{\mathcal{M}}} \mathbb{E}^{\sigma}_{\mathcal{M},v}(\varphi)$. For $\varepsilon > 0$, a strategy σ is *ε-optimal* for objective φ if $\mathbb{E}^{\sigma}_{\mathcal{M},v}(\varphi) \geq \mathbb{E}_{\mathcal{M},v}(\varphi) - \varepsilon$. A strategy σ is *optimal* for objective φ if it achieves the φ-value of the vertex, that is, if $\mathbb{E}^{\sigma}_{\mathcal{M},v}(\varphi) = \mathbb{E}_{\mathcal{M},v}(\varphi)$.

Two-player games. An MDP $\mathcal{M} = ((V,E),(V_\circ,V_\diamond),\mathbb{P},w)$ can be seen as a two-player game, denoted $\mathcal{G}_{\mathcal{M}} = ((V,E),(V_\circ,V_\diamond),w)$ where the probability vertices in $V_\diamond$ are interpreted as vertices belonging to an adversarial environment, the probability function $\mathbb{P}$ is forgotten, and the adversary chooses a strategy of its choice. Using both interpretations, which are that of a stochastic model as well as the adversarial environment, is crucial to our work.

Maximal end components. An *end component* (EC) in an MDP is a subset $T \subseteq V$ of vertices such that for every probabilistic vertex v in T, every out-neighbour of v belongs to the subset T, and T is strongly connected. Thus, for every pair of vertices v, v' in the subset T, the player has a strategy to reach v' from v with probability 1 and the player can ensure with probability 1 that the token never leaves T. A *maximal end component* (MEC) is an EC that is not contained in any other EC. The MECs in an MDP are disjoint. Each vertex in an MDP belongs to either no MEC or exactly one MEC. Thus, the number of MECs is bounded above by the number of vertices in the MDP. We denote by $\mathfrak{M}$ the set of MECs of $\mathcal{M}$. The MEC decomposition can computed in polynomial time [13]. We now recall some classical results on MDPs.

Lemma 2.1 (Optimal reachability [3]). *Given an MDP $\mathcal{M}$ and a set $T \subseteq V$ of target states, we can compute in polynomial time for each vertex $v \in V$, the probability $p^*_v = \sup_\sigma \Pr^{\sigma}_{\mathcal{M},v}(\mathsf{Reach}(T))$ with which the player can ensure visiting T. There is an optimal pure memoryless strategy σ^* that enforces reaching T with probability p^*_v from every vertex $v \in V$. Further, for all $v \in V, c < p^*_v$, there exists $N \in \mathbb{N}$ such that by playing σ^* for N steps, we reach T from v with probability greater than c.*

For an arbitrary strategy σ of the player, it is almost-surely the case that an outcome ends up in an MEC of $\mathcal{M}$.

Lemma 2.2 (Long-run appearance in MECs [3]). *Given an MDP $\mathcal{M}$ with a set V of vertices, for every strategy σ of the player and for every vertex $v \in V$, we have that $\sum_{M\in\mathfrak{M}} \Pr^{\sigma}_{\mathcal{M},v}(\inf(\pi) \subseteq M) = 1$.*

Window mean-payoff objectives. In this work, we look at the BWC framework in the context of the window mean-payoff objectives. We first define Boolean versions of the objective.

For a run $\pi = v_0v_1v_2\cdots$ in an MDP $\mathcal{M}$, 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 $\lambda \in \mathbb{Q}$, a run $\pi = v_0v_1\cdots$ in $\mathcal{M}$ satisfies the *fixed window mean-payoff objective* $\mathsf{FWMP}_{\mathcal{M}}(\ell,\lambda)$ 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.

$$\mathsf{FWMP}_{\mathcal{M}}(\ell,\lambda) = \{\pi \in \mathsf{Runs}^{\mathcal{M}} \mid \exists k \geq 0 \cdot \forall i \geq k \cdot \exists j \in \{1,\dots,\ell\} : \mathsf{MP}(\pi(i,i+j)) \geq \lambda\}$$

Corresponding to the Boolean objective $\mathsf{FWMP}_{\mathcal{M}}(\ell,\lambda)$, we define a quantitative version of the objective as follows: Given a run π in an MDP $\mathcal{M}$, the $\varphi_{\mathsf{FWMP}(\ell)}$-value of π is equal to $\sup\{\lambda \in \mathbb{R} \mid \pi \in \mathsf{FWMP}(\ell,\lambda)\}$, the supremum threshold λ such that the run satisfies $\mathsf{FWMP}_{\mathcal{M}}(\ell,\lambda)$. For a run π in an MDP, the $\varphi_{\mathsf{FWMP}(\ell)}$-value of π can be of the form $\frac{a}{b}$ where $a \in \{-W\cdot\ell,\dots,0,\dots,W\cdot\ell\}$ and $b \in \{1,\dots,\ell\}$. Hence the $\varphi_{\mathsf{FWMP}(\ell)}$-value can be one of finitely many values leading to the following.

Proposition 2.3. *For all $\ell \geq 1$ and all $\lambda \in \mathbb{R}$, we have $\pi \in \{\varphi_{\mathsf{FWMP}(\ell)} \geq \lambda\}$ if and only if $\pi \in \mathsf{FWMP}(\ell,\lambda)$.*

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

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

We define the φ_{BWMP}-value of a run π analogously to $\varphi_{\mathsf{FWMP}(\ell)}$. Given a run π in an MDP $\mathcal{M}$, the φ_{BWMP}-value of π is equal to $\sup\{\lambda \in \mathbb{R} \mid \pi \in \mathsf{BWMP}(\lambda)\}$, or equivalently, $\sup\{\lambda \in \mathbb{R} \mid \exists \ell \geq 1 : \pi \in \mathsf{FWMP}(\ell,\lambda)\}$.

An observation similar to Proposition 2.3 does not hold for the bounded window mean-payoff objective. This is because since ℓ can be unbounded, there may be a run π such that π does not satisfy $\mathsf{FWMP}(\ell,\varphi_{\mathsf{BWMP}}(\pi))$ for any $\ell \geq 1$. However, the following holds.

Proposition 2.4. *For all $\lambda \in \mathbb{R}$, if $\pi \in \mathsf{BWMP}(\lambda)$, then $\pi \in \{\varphi_{\mathsf{BWMP}} \geq \lambda\}$.*

Note that both $\mathsf{FWMP}_{\mathcal{M}}(\ell,\lambda)$ and $\mathsf{BWMP}_{\mathcal{M}}(\lambda)$ are Boolean prefix-independent objectives. We omit the subscript $\mathcal{M}$ when it is clear from the context. As considered in previous works [7, 9, 11], the window length ℓ is usually small (typically $\ell \leq |V|$), and therefore we assume that ℓ is given in unary (while the edge-payoffs are given in binary).

3 PROBLEM DEFINITION

We formally describe the notion of optimising expected φ-value with guarantees. Given an MDP $\mathcal{M}$, a vertex v, a guarantee threshold α, and an expectation threshold β, we consider the following decision problems for optimising expectation with guarantees.

(1) **Beyond worst-case (BWC) synthesis [10]** (Expectation maximisation with sure guarantee): The problem here is to check if the supremum of $\mathbb{E}^{\sigma}_{\mathcal{M},v}(\varphi)$ over all strategies σ such that $\mathsf{Out}^{\mathcal{M}}_v(\sigma) \subseteq \{\varphi \geq \alpha\}$ (that is, all outcomes in $\mathcal{M}$ starting from v that are consistent with σ have φ-value at least α) is at least β. We write this decision problem succinctly as $v \models \mathsf{BWC}(\alpha,\beta)$ in MDP $\mathcal{M}$ for objective φ. Note that for the $\mathsf{Out}^{\mathcal{M}}_v(\sigma) \subseteq \{\varphi \geq \alpha\}$ part, the probabilities are ignored and the environment is considered antagonistic in the sense that every play consistent with strategy σ needs to satisfy the threshold Boolean constraint $\{\varphi \geq \alpha\}$.

(2) **Beyond probability threshold (BPT) synthesis [15]** (Expectation maximisation with probabilistic guarantee): Here we are given an additional probabilistic threshold p. The problem here is to check if the supremum of $\mathbb{E}^{\sigma}_{\mathcal{M},v}(\varphi)$ over all strategies σ such that $\Pr^{\sigma}_{\mathcal{M},v}(\{\varphi \geq \alpha\}) \geq p$ is at least β. We write this decision problem succinctly as $v \models \mathsf{BPT}((p,\alpha),\beta)$ in MDP $\mathcal{M}$ for objective φ.

(3) **Beyond almost-sure (BAS) synthesis [18]** (Expectation maximisation with almost-sure guarantee): The problem here is to check if the supremum of $\mathbb{E}^{\sigma}_{\mathcal{M},v}(\varphi)$ over all strategies σ such that $\Pr^{\sigma}_{\mathcal{M},v}(\{\varphi \geq \alpha\}) = 1$ is at least β. We write this decision problem succinctly as $v \models \mathsf{BAS}(\alpha, \beta)$ in MDP $\mathcal{M}$ for objective φ.

For each of these decision problems, we study the case where φ is either $\varphi_{\mathsf{FWMP}(\ell)}$ or φ_{BWMP}. For the BWC synthesis problem, if the answer is yes, then for every $\varepsilon > 0$, we construct a strategy that achieves an expected φ-value of at least $\beta - \varepsilon$. For the BPT and the BAS synthesis problems, if the answer is yes, then we construct strategies that achieve the expected φ-value of at least β as well as the specified guarantees.

For all three synthesis problems for the classical mean-payoff objective and for the window mean-payoff objectives, we can assume without loss of generality that the guarantee threshold α is 0. This is because we have $v \models \mathsf{BWC}(\alpha, \beta)$ in an MDP $\mathcal{M}$ if and only if $v \models \mathsf{BWC}(0, \beta - \alpha)$ in a new MDP $\mathcal{M}_{-\alpha}$ (obtained from $\mathcal{M}$ by subtracting α from every edge payoff in $\mathcal{M}$). Similarly, we can set $\alpha = 0$ for BPT and BAS without loss of generality.

4 EXPECTED FIXED WINDOW MEAN-PAYOFF VALUE WITH GUARANTEES

In this section, we show that the BWC, BPT, and the BAS synthesis problems for $\varphi_{\mathsf{FWMP}(\ell)}$ can be solved with no additional complexity than that of either of the special cases: maximising the expectation without any guarantee, or ensuring guarantee surely, almost-surely, or with a certain probability while disregarding any expected performance.

4.1 Sure Guarantee

Recall that the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of a vertex v is defined as the supremum of the expected $\varphi_{\mathsf{FWMP}(\ell)}$-values $\mathbb{E}^{\sigma}_{\mathcal{M},v}(\varphi_{\mathsf{FWMP}(\ell)})$ over all strategies σ of the player. We show in Example 4.1 that in general, an optimal strategy achieving the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value β while also ensuring the sure guarantee of 0 may not exist, but for every $\varepsilon > 0$, an ε-optimal strategy exists. We thus show how to construct ε-optimal strategies for $\mathsf{BWC}(0, \beta)$ satisfaction.

Example 4.1. Consider the MDP shown in Figure 1. We want to determine if $v_2 \models \mathsf{BWC}(0, 2)$ for the $\varphi_{\mathsf{FWMP}(\ell)}$ objective for window length $\ell = 3$. If the token somehow reaches v_4, then from there, the player has a strategy to ensure that the $\varphi_{\mathsf{FWMP}(\ell)}$-value of the outcome is surely 2, and thus, the player can ensure that the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value from v_4 is at least 2 as well. Note that for every successive visit of the token to v_7, the player has to alternate between v_6 and v_8 to ensure that the outcome is 2. However, starting from v_2, the player does not have a strategy to reach v_4 surely. If the token remains in the set $\{v_0, v_1, v_2\}$, then the $\varphi_{\mathsf{FWMP}(\ell)}$-value that can be surely attained is 0. However, if the player tries to move the token from v_2 to v_3 some fixed (but large) number N of times, then the probability of reaching v_4 can be made close to 1. If after N tries, the token does not reach v_4, then the player can choose to keep the token in the set $\{v_0, v_1, v_2\}$ and thus surely get a $\varphi_{\mathsf{FWMP}(\ell)}$-value of 0. This gives a strategy that surely ensures that the $\varphi_{\mathsf{FWMP}(\ell)}$-value of the outcome is non-negative and the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value is at least $2 - \varepsilon$ for all $\varepsilon > 0$.

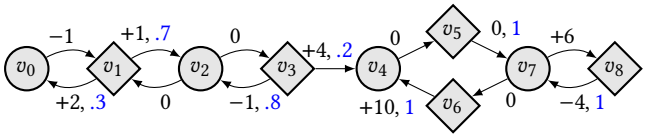

Figure 1: An example of an MDP for BWC with $\ell = 3$.

We present an algorithm (Algorithm 1) that, given a vertex v_{init} in an MDP $\mathcal{M}_0$, and a threshold β, decides if $v_{\mathsf{init}} \models \mathsf{BWC}(0, \beta)$. We show that it runs in time that is polynomial in the size of the input, and explain how it yields an ε-optimal strategy for the player.

Algorithm 1 BWC synthesis for $\varphi_{\mathsf{FWMP}(\ell)}$ objective

Input: MDP $\mathcal{M}_0$, vertex $v_{\mathsf{init}} \in V$, window length ℓ, and expectation threshold β

Output: Yes if and only if $v_{\mathsf{init}} \models \mathsf{BWC}(0, \beta)$

1: Compute $W^{\mathsf{FWMP}(\ell)}_{\mathsf{S},0}$, the sure winning region in $\mathcal{M}_0$ for objective $\{\varphi_{\mathsf{FWMP}(\ell)} \geq 0\}$.
2: **if** $v_{\mathsf{init}} \notin W^{\mathsf{FWMP}(\ell)}_{\mathsf{S},0}$ **then return** No
3: **if** $\beta \leq 0$ **then return** Yes
4: Construct $\mathcal{M}_1 := \mathcal{M}_0 \restriction W^{\mathsf{FWMP}(\ell)}_{\mathsf{S},0}$, the MDP obtained by restricting $\mathcal{M}_0$ to $W^{\mathsf{FWMP}(\ell)}_{\mathsf{S},0}$.
5: **for** $v \in V_{\circ}$ in $\mathcal{M}_1$ **do**
6: Compute the maximum λ_v such that v belongs to the sure winning region in $\mathcal{M}_1$ for objective $\{\varphi_{\mathsf{FWMP}(\ell)} \geq \lambda_v\}$.
7: Construct $\mathcal{M}_2$ from $\mathcal{M}_1$ as follows:
Change payoffs of all edges to -1.
For each player vertex v, add a self-loop with payoff λ_v.
8: **return** Yes if and only if $\mathbb{E}_{\mathcal{M}_2, v_{\mathsf{init}}}(\varphi_{\mathsf{MP}}) \geq \beta$.

Description of Algorithm 1. We first compute the sure winning region, $W^{\mathsf{FWMP}(\ell)}_{\mathsf{S},0}$, of the player for the threshold objective $\{\varphi_{\mathsf{FWMP}(\ell)} \geq 0\}$ (Line 1). The sure winning region of the player in MDP $\mathcal{M}_0$ is the same as the winning region of the player in the adversarial two-player game $\mathcal{G}_{\mathcal{M}_0}$ obtained by viewing probabilistic vertices of $\mathcal{M}_0$ as vertices of an adversary. We compute the winning region of player in $\mathcal{G}_{\mathcal{M}_0}$ using [11, Algorithm 1]. If the vertex v_{init} does not belong to $W^{\mathsf{FWMP}(\ell)}_{\mathsf{S},0}$, then the sure guarantee cannot be satisfied from v_{init}, and we have that $v_{\mathsf{init}} \not\models \mathsf{BWC}(0, \beta)$ for all $\beta \in \mathbb{Q}$, and the algorithm returns No.

Otherwise, we have that $v_{\mathsf{init}} \in W^{\mathsf{FWMP}(\ell)}_{\mathsf{S},0}$, and thus, there exists a strategy that ensures the sure satisfaction of $\{\varphi_{\mathsf{FWMP}(\ell)} \geq 0\}$ from v_{init}. We now check if $\beta \leq 0$. This is because the sure satisfaction of $\{\varphi_{\mathsf{FWMP}(\ell)} \geq 0\}$ from v_{init} implies that the expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of v_{init} is at least 0, and in particular, if $\beta \leq 0$, then it follows that $v_{\mathsf{init}} \models \mathsf{BWC}(0, \beta)$, and the algorithm returns Yes.

Finally, we arrive at the interesting case that is $\beta > 0$. From every vertex v in $W^{\mathsf{FWMP}(\ell)}_{\mathsf{S},0}$, there exists a sure winning strategy for objective $\{\varphi_{\mathsf{FWMP}(\ell)} \geq 0\}$, and every such sure winning strategy never moves the token out of $W^{\mathsf{FWMP}(\ell)}_{\mathsf{S},0}$. Thus, we can prune all vertices from $\mathcal{M}_0$ that are not in $W^{\mathsf{FWMP}(\ell)}_{\mathsf{S},0}$ to obtain $\mathcal{M}_1$, and we

have that the sets of sure winning strategies for $\{\varphi_{\mathsf{FWMP}(\ell)} \geq 0\}$ in $\mathcal{M}_0$ and $\mathcal{M}_1$ are the same. Next, in Line 6, for each player vertex v in $\mathcal{M}_1$, we compute λ_v, that is the maximum $\varphi_{\mathsf{FWMP}(\ell)}$-value that the player can surely ensure in $\mathcal{M}_1$ starting from v. The sure $\varphi_{\mathsf{FWMP}(\ell)}$-value of each vertex can be computed in polynomial time using binary search (details appear later). We construct a new MDP $\mathcal{M}_2$ which has the same set of vertices as $\mathcal{M}_1$. Each edge in $\mathcal{M}_1$ is also present in $\mathcal{M}_2$ but with payoff -1. In addition, for each player vertex v in $\mathcal{M}_2$, we add a self-loop with payoff λ_v to v. We compute the expected φ_{MP}-value of v_{init} in $\mathcal{M}_2$ using linear programming [25]. The correctness of Line 8 follows from Lemma 4.2.

Lemma 4.2. *For all $\gamma \geq 0$, we have that $v_{init} \models \mathsf{BWC}(0, \gamma)$ in $\mathcal{M}_0$ if and only if $\mathbb{E}_{\mathcal{M}_2, v_{init}}(\varphi_{\mathsf{MP}}) \geq \gamma$.*

Proof sketch. Let σ_{MP} be a deterministic memoryless optimal strategy from v_{init} in $\mathcal{M}_2$ for the expectation of φ_{MP} (the existence of such a strategy follows from [25]). We prove that for every $\varepsilon > 0$, we can construct a strategy σ^*_ε such that outcomes of this strategy from v_{init} surely satisfy $\{\varphi_{\mathsf{FWMP}(\ell)} \geq 0\}$ and we also have $\mathbb{E}^{\sigma^*_\varepsilon}_{\mathcal{M}, v_{\mathsf{init}}}(\varphi_{\mathsf{FWMP}(\ell)}) \geq \gamma - \varepsilon$.

Each run π from v_{init} that is an outcome of σ_{MP} almost-surely eventually reaches a vertex u from which it always takes the self-loop on u with edge payoff λ_u. This is because σ_{MP} is memoryless and the payoff λ_u of the self-loop for every vertex u is non-negative, while all other edges in $\mathcal{M}_2$ have a negative payoff of -1. Let $u_1, \ldots, u_k$ be the vertices to which an outcome of σ_{MP} reaches with positive probabilities $p_1, \ldots, p_k$ respectively upon reaching which the token starts looping. From Lemma 2.1, for every $\varepsilon > 0$, we can choose a large enough N such that the token reaches each u_i with probability at least $p_i - \varepsilon/(|V| \cdot W)$, where W is the maximum edge payoff appearing in $\mathcal{M}_0$. The strategy σ^*_ε mimics σ_{MP} until σ_{MP} starts looping or until N steps have passed, whichever comes first. Then, if the token is on some vertex v, then σ^*_ε switches to mimicking the sure-winning strategy $\sigma_{\mathsf{S}}^{\mathsf{FWMP}(\ell)}$ from v for the rest of the run. It follows that the strategy σ^*_ε will reach one of the u_i vertices with probability $\sum_{i=1}^{k}(p_i - \varepsilon/(|V| \cdot W))$ and that expected $\varphi_{\mathsf{FWMP}(\ell)}$-value of an outcome of σ^*_ε is at least $\gamma - \varepsilon$. Moreover, the $\varphi_{\mathsf{FWMP}(\ell)}$-value of an outcome of σ^*_ε is surely non-negative since $\lambda_v \geq 0$ for all vertices v in $\mathcal{M}_1$. Thus, we have that $v_{\mathsf{init}} \models \mathsf{BWC}(0, \gamma)$.

For the converse, suppose that from vertex v_{init}, the player has an ε-optimal strategy σ^*_ε for $\mathsf{BWC}(0, \gamma)$ in $\mathcal{M}_0$. Using σ^*_ε, we construct a strategy σ_{MP} from v_{init} in $\mathcal{M}_2$ that achieves expected φ_{MP}-value at least γ, that is, $\mathbb{E}^{\sigma_{\mathsf{MP}}}_{\mathcal{M}_2, v_{\mathsf{init}}}(\varphi_{\mathsf{MP}}) \geq \gamma$. From Lemma 2.2, we have that an outcome of σ^*_ε almost-surely eventually reaches and stays in an MEC from which it never exits. Suppose that starting from v_{init}, an outcome of the strategy σ^*_ε ends up in MECs $M_1, M_2, \ldots, M_k$ with probability $p_1, p_2, \ldots, p_k$ respectively. If in an outcome π of σ^*_ε, the token reaches the MEC M_i and never leaves, then the $\varphi_{\mathsf{FWMP}(\ell)}$-value of π is at most $\max\{\lambda_v \mid v \in M_i\}$, and we denote this by λ_{M_i}. The strategy σ_{MP} mimics σ^*_ε to almost-surely reach the same MECs in $\mathcal{M}_2$ with the same probabilities. In each MEC M in $\mathcal{M}_2$, the strategy σ_{MP} can ensure expected φ_{MP}-value $\max\{\lambda_v \mid v \in M\}$ by almost-surely reaching the vertex v in M with the maximum λ_v, and then looping on v forever. Thus, we have that $\mathbb{E}_{\mathcal{M}_2, v_{\mathsf{init}}}(\varphi_{\mathsf{MP}}) \geq \gamma - \varepsilon$. Since this holds for every ε, we have that $\mathbb{E}_{\mathcal{M}_2, v_{\mathsf{init}}}(\varphi_{\mathsf{MP}}) \geq \gamma$. □

Memory requirement for ε-optimal strategies. In the proof sketch above, the strategy $\sigma_{\mathsf{S}}^{\mathsf{FWMP}(\ell)}$ requires at most ℓ memory [11]. From Lemma 2.1, for all $\varepsilon > 0$, there exists an integer N that the Mealy machine stores in its state space. Thus, deterministic finite-memory strategies suffice for BWC. It suffices to consider N logarithmic in $1/\varepsilon$. This bound is obtained by solving a linear recurrence relation involving the transition probabilities in the arena [21].

Running time analysis. Given a vertex v in an MDP $\mathcal{M}_0$ and a threshold α, the problem of determining if the player has a strategy to surely satisfy the objective $\{\varphi_{\mathsf{FWMP}(\ell)} \geq \alpha\}$ from v is in polynomial time [11]. Thus, the set $W_{\mathsf{S},0}^{\mathsf{FWMP}(\ell)}$ of all such vertices can be computed in polynomial time.

Computing the sure value λ_v for the $\varphi_{\mathsf{FWMP}(\ell)}$ for each vertex can be done in polynomial time by using binary search. Since λ_v is non-negative, recall that we can write λ_v as a fraction a/b, where $b \in \{1, \ldots, \ell\}$ and $a \in \{0, 1, \ldots, W \cdot \ell\}$. Thus, there are at most $W \cdot \ell^2$ different values that λ_v can take. For each possible value α, we check if v belongs to the sure winning region in $\mathcal{M}_1$ for the threshold objective $\{\varphi_{\mathsf{FWMP}(\ell)} \geq \alpha\}$. Since λ_v takes at most $W \cdot \ell^2$ different values, it takes at most $\log(W \cdot \ell^2)$ checks to arrive at λ_v and thus λ_v can be computed in time that is polynomial in the size of the input.

Expectation of φ_{MP} objective can also be solved in polynomial time using linear programming [25]. Thus, Algorithm 1 runs in polynomial time. The following theorem summarises our results.

Theorem 4.3. BWC *synthesis for $\varphi_{\mathsf{FWMP}(\ell)}$ with ℓ given in unary is in* PTIME*, and if $v \models \mathsf{BWC}(\alpha, \beta)$, then for every $\varepsilon > 0$, there exists an ε-optimal finite-memory deterministic strategy from v.*

4.2 Probabilistic Guarantee

Next, we look at the $\mathsf{BPT}((p, \alpha), \beta)$ synthesis problem. As before, we assume without loss of generality that α is equal to zero. In contrast to BWC, in the case of BPT, the problem is interesting even when $\beta \leq 0$. This is because satisfying the threshold objective $\{\varphi_{\mathsf{FWMP}(\ell)} \geq 0\}$ with probability at least p does not necessarily imply that the expectation is at least β, even when $\beta \leq 0$. Further, unlike in the case of BWC, we cannot prune away the set of vertices from which the player cannot satisfy $\{\varphi_{\mathsf{FWMP}(\ell)} \geq 0\}$ with probability at least p. This is because, in trying to satisfy the expectation threshold, the token may end up visiting a vertex from which the probability of satisfying $\{\varphi_{\mathsf{FWMP}(\ell)} \geq 0\}$ is less than p.

Example 4.4. In Figure 2, we see an MDP $\mathcal{M}$ in which we want to determine if $v_3 \models \mathsf{BPT}((0.5, 0), 2)$ for $\varphi_{\mathsf{FWMP}(\ell)}$ for window length $\ell = 2$. If the player keeps the token in the MEC consisting of $\{v_0, v_1, v_2, v_3\}$ forever with probability 1, then a $\varphi_{\mathsf{FWMP}(\ell)}$-value of $+1$ (which is non-negative) is ensured with probability 1, which satisfies the guarantee threshold. However, this is not sufficient to satisfy the expectation threshold 2. On the other hand, if the player moves the token from v_3 to v_4 with probability 1, then the token reaches the MEC $\{v_5, v_7\}$ with probability 0.6 and achieves a value of -1, whereas the token reaches the MEC $\{v_6, v_8\}$ with probability 0.4 which has a value of 9. Thus, from v_4, the expected value is $0.6 \cdot (-1) + 0.4 \cdot 9 = 3$ and the expectation threshold is satisfied. However, the token achieves non-negative value with probability

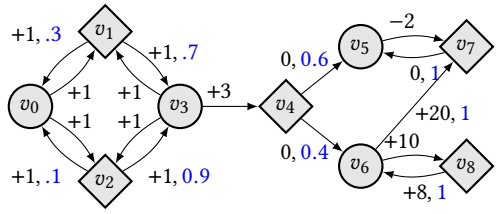

Figure 2: An example of an MDP for $\mathsf{BPT}((0.5, 0), 2)$ with $\ell = 2$.

only 0.4, and the guarantee threshold is not satisfied with sufficient probability. Thus, a strategy satisfying both expectation and probabilistic guarantee requires randomisation, and in fact, we can show that if the randomised strategy is to stay in $\{v_0, v_1, v_2, v_3\}$ with probability q and to eventually go to v_4 with probability $1 - q$, then this strategy satisfies BPT for $\frac{1}{6} \leq q \leq \frac{1}{3}$. This shows that deterministic strategies are not sufficient for BPT synthesis and that randomised strategies are strictly more powerful. Moreover, in order to always stay in the MEC $\{v_0, v_1, v_2, v_3\}$ with probability $\frac{1}{4}$ (which is strictly between 0 and 1), the strategy needs memory.

Algorithm 2 BPT synthesis for $\varphi_{\mathsf{FWMP}(\ell)}$ objective

Input: MDP $\mathcal{M}_0$, vertex $v_{\mathsf{init}} \in V$, window length ℓ, probabilistic-guarantee threshold $(p, 0)$, and expectation threshold β
Output: Yes if and only if $v \models \mathsf{BPT}((p, 0), \beta)$

1: Compute MEC decomposition $\mathfrak{M}$ of $\mathcal{M}_0$.
2: **for** $M \in \mathfrak{M}$ in $\mathcal{M}_0$ **do**
3: Compute the maximum μ_M such that the player almost-surely satisfies the threshold objective $\{\varphi_{\mathsf{FWMP}(\ell)} \geq \mu_M\}$ from every vertex v in the MDP $\mathcal{M}_0$ restricted to the MEC M.
4: Construct $\mathcal{M}_1$ from $\mathcal{M}_0$ as follows:
Collapse each MEC M into a player vertex v_M, and add to it a self-loop with payoff μ_M.
5: **return** Yes if and only if $v_{\mathsf{init}} \models \mathsf{BPT}((p, 0), \beta)$ in $\mathcal{M}_1$ for the φ_{MP} objective.

Description of Algorithm 2. We begin by finding the MEC decomposition $\mathfrak{M}$ of $\mathcal{M}_0$. Then, for every MEC $M \in \mathfrak{M}$, we compute the maximum $\varphi_{\mathsf{FWMP}(\ell)}$-value μ_M that can be achieved almost-surely from a vertex in the MEC M. This is well-defined as all vertices in a MEC have the same value since $\varphi_{\mathsf{FWMP}(\ell)}$ is a prefix-independent objective and every vertex in a MEC is almost-surely reachable from every other vertex in the MEC. We then construct a new MDP $\mathcal{M}_1$ by collapsing each MEC M in $\mathcal{M}_0$ to a vertex v_M. Finally, for each collapsed MEC M in $\mathcal{M}_1$, we add a self-loop with payoff μ_M. In this collapsed MDP $\mathcal{M}_1$, we solve the BPT problem for the φ_{MP} objective, that is, classical mean-payoff objective, by solving a linear program as in [15]. The LP L we construct is simpler than in [15] since each MEC in the collapsed MEC has only one player vertex and a probabilistic vertex. If the vertex v_{init} does not belong to any MEC in $\mathcal{M}_0$, then it appears in the collapsed MDP $\mathcal{M}_1$ as well. Otherwise, if v_{init} belongs to some MEC M_{init} in $\mathcal{M}_0$, then for ease of notation, we continue to use v_{init} to represent the player vertex in $\mathcal{M}_1$ obtained after collapsing M_{init}.

Lemma 4.5. *We have $v_{init} \models \mathsf{BPT}((p, 0), \beta)$ in $\mathcal{M}_0$ for $\varphi_{\mathsf{FWMP}(\ell)}$ if and only if $v_{init} \models \mathsf{BPT}((p, 0), \beta)$ in $\mathcal{M}_1$ for φ_{MP}.*

Proof sketch. An optimal strategy σ_{MP} for $\mathsf{BPT}((p, 0), \beta)$ for φ_{MP} is one that tries to reach MECs with different probabilities, and then at some point, starts looping on those MECs. Using σ_{MP}, we construct a strategy $\sigma_{\mathsf{FWMP}(\ell)}$ that is optimal for $\mathsf{BPT}((p, 0), \beta)$ for $\varphi_{\mathsf{FWMP}(\ell)}$ in $\mathcal{M}_0$. The strategy $\sigma_{\mathsf{FWMP}(\ell)}$ mimics the strategy σ_{MP} in $\mathcal{M}_0$ until σ_{MP} switches to looping. When σ_{MP} reaches an MEC and switches to looping there, the strategy $\sigma_{\mathsf{FWMP}(\ell)}$ switches to $\sigma^{\mathsf{AS}}_{\mathsf{FWMP}(\ell)}$, an almost-sure winning strategy for $\{\varphi_{\mathsf{FWMP}(\ell)} \geq \mu_M\}$. If the switch happens at a vertex in an MEC M in $\mathcal{M}_0$, then subsequently, the value of the run is μ_M almost-surely. Thus, if threshold 0 is achieved with probability p in $\mathcal{M}_1$, then the same threshold is achieved with the same probability in original MDP $\mathcal{M}_0$, and if an expectation value β is attained in $\mathcal{M}_1$, then the same expectation value β is also achieved in $\mathcal{M}_0$.

For the converse, given an optimal strategy $\sigma_{\mathsf{FWMP}(\ell)}$ for the $\mathsf{BPT}((p, 0), \beta)$ synthesis problem for $\varphi_{\mathsf{FWMP}(\ell)}$ in $\mathcal{M}_0$, we construct an optimal strategy σ_{MP} for $\mathsf{BPT}((p, 0), \beta)$ for φ_{MP} in $\mathcal{M}_1$. By Lemma 2.2, we have that playing according to the strategy $\sigma_{\mathsf{FWMP}(\ell)}$, the token eventually moves into an MEC M from which it never exits. The strategy σ_{MP} mimics $\sigma_{\mathsf{FWMP}(\ell)}$ up to this point, after which σ_{MP} switches to looping on v_M. One can see that if the expectation and probability threshold are satisfied in $\mathcal{M}_0$ for $\varphi_{\mathsf{FWMP}(\ell)}$, then they are also satisfied in $\mathcal{M}_1$ for φ_{MP}. □

Running time analysis. The MEC decomposition of $\mathcal{M}_0$ can be done in polynomial time [13]. For each MEC M in $\mathcal{M}_0$, computing the value of μ_M can also be done in polynomial time using binary search [7]. Finally, to check if $v \models \mathsf{BPT}((p, 0), \beta)$ in $\mathcal{M}_1$ for the φ_{MP}, the algorithm solves the linear program L, which can be done in polynomial time since the number of variables and constraints in L is polynomial in the size of the input [24].

Memory requirements of optimal strategies. Figure 2 shows that, in general, optimal strategies of the player requires memory and randomisation, and thus, deterministic strategies do not suffice. However, finite memory suffices. In particular, before reaching the MECs in $\mathcal{M}_0$, the strategy $\sigma_{\mathsf{FWMP}(\ell)}$ may need memory as well as randomisation as illustrated in Example 4.4, and the strategy $\sigma^{\mathsf{AS}}_{\mathsf{FWMP}(\ell)}$ can be a deterministic strategy with memory size ℓ [19].

Theorem 4.6. *BPT synthesis for $\varphi_{\mathsf{FWMP}(\ell)}$ objective with ℓ given in unary is in* PTIME, *and if $v \models \mathsf{BPT}((p, \alpha), \beta)$, then there exists an optimal finite-memory randomised strategy from v.*

4.3 Almost-sure Guarantee

In this section, we solve the BAS synthesis problem, that is, we decide, given an MDP $\mathcal{M}_0$, a vertex v_{init}, and an expectation threshold β, if v_{init} satisfies $\mathsf{BAS}(0, \beta)$ in $\mathcal{M}_0$. Similar to the BWC synthesis problem, we are interested in the case when $\beta > 0$.

The $\mathsf{BAS}(0, \beta)$ synthesis problem is a special case of $\mathsf{BPT}((p, 0), \beta)$ when $p = 1$. We get better bounds for the memory size of the optimal strategies for BAS as compared to BPT in general, and moreover, deterministic strategies suffice for BAS. Note that Algorithm 2 works for $\mathsf{BAS}(0, \beta)$ as well if we let $p = 1$. In Line 5 in Algorithm 2, we need to check if $v_{\mathsf{init}} \models \mathsf{BPT}((1, 0), \beta)$ in the collapsed MDP $\mathcal{M}_1$ for

the φ_{MP} objective. That is, we need to check if $v_{\mathsf{init}} \models \mathsf{BAS}(0,\beta)$ in $\mathcal{M}_1$ for φ_{MP}. Instead of using the linear program as described in Section 4.2 to do the check, we use a reduction of the BAS problem for φ_{MP} to the problem of standard expected φ_{MP}-value that is described in [15]. The reduction is as follows: We prune from $\mathcal{M}_1$ all vertices from which the player cannot almost-surely achieve non-negative φ_{MP}-value to get an MDP $\mathcal{M}_2$. By analysing the MECs, we can check in PTIME if the player can almost-surely achieve non-negative φ_{MP}-value from a vertex. If v_{init} is pruned away, then it does not satisfy $\mathsf{BAS}(0,\beta)$ for φ_{MP} in $\mathcal{M}_1$. Otherwise, v_{init} satisfies $\mathsf{BAS}(0,\beta)$ in $\mathcal{M}_1$ if and only if in the pruned MDP $\mathcal{M}_2$, the expected φ_{MP}-value of v_{init} in $\mathcal{M}_2$ is at least β. The correctness of the algorithm follows from Lemma 4.5 by setting $p = 1$.

Memory requirements of optimal strategies. Let σ_{MP} be a memoryless deterministic optimal strategy for expected classical mean-payoff objective φ_{MP} in $\mathcal{M}_2$. An optimal strategy σ^* for $\mathsf{BAS}(0,\beta)$ for $\varphi_{\mathsf{FWMP}(\ell)}$ in $\mathcal{M}_0$ can be constructed by first mimicking σ_{MP} until σ_{MP} switches to looping. If the token is on a vertex v when this switch happens in $\mathcal{M}_2$, then σ^* should switch to mimicking an optimal strategy for the almost-sure satisfaction of the threshold objective $\{\varphi_{\mathsf{FWMP}(\ell)} \geq \mu_M\}$ in $\mathcal{M}_0$.

Since there exist deterministic memoryless optimal strategies for expectation of φ_{MP} [25], and there exist deterministic optimal strategies with memory size at most ℓ for the almost-sure satisfaction of $\{\varphi_{\mathsf{FWMP}(\ell)} \geq \mu_M\}$ [19], we get that there exist deterministic optimal strategies with memory size at most ℓ for $\mathsf{BAS}(0,\beta)$.

Theorem 4.7. BAS *synthesis for* $\varphi_{\mathsf{FWMP}(\ell)}$ *objective with* ℓ *given in unary is in* PTIME*, and if* $v \models \mathsf{BAS}(\alpha,\beta)$*, then there exists an optimal deterministic strategy of memory size at most* ℓ *from* v*.*

5 EXPECTED BOUNDED WINDOW MEAN-PAYOFF VALUE WITH GUARANTEES

In this section, we study the expectation maximisation problem with sure, almost-sure, and probabilistic guarantees for the bounded window mean-payoff objective. The algorithms are similar to those for the fixed window mean-payoff objective described in the previous section. We only highlight the main differences here. Note that all our algorithms require solving two-player games with either $\{\varphi_{\mathsf{BWMP}} \geq 0\}$ or $\{\varphi_{\mathsf{MP}} \geq 0\}$ objective. While two-player games with $\{\varphi_{\mathsf{FWMP}(\ell)} \geq 0\}$ objective can be solved in PTIME, solving two-player games with $\{\varphi_{\mathsf{BWMP}} \geq 0\}$ objective or $\{\varphi_{\mathsf{MP}} \geq 0\}$ objective is in NP ∩ coNP [7].

Sure guarantee. The algorithm here is similar to Algorithm 1. As stated above, computing $W^{\mathsf{BWMP}}_{\mathsf{S},0}$ is in NP ∩ coNP. The MDP $\mathcal{M}_1 := \mathcal{M}_0 \restriction W^{\mathsf{BWMP}}_{\mathsf{S},0}$ is the MDP obtained by restricting $\mathcal{M}_0$ to $W^{\mathsf{BWMP}}_{\mathsf{S},0}$. For every $v \in V_\circ$, we compute the maximum λ_v such that v belongs to the sure winning region in $\mathcal{M}_1$ for the $\{\varphi_{\mathsf{BWMP}} \geq \lambda_v\}$ objective. Recall that for a run π, we have $\varphi_{\mathsf{BWMP}}(\pi) = \sup\{\lambda \in \mathbb{R} \mid \exists \ell \geq 0 : \pi \in \mathsf{FWMP}(\ell,\lambda)\}$. The maximum λ_v can be computed by solving the two-player game $\mathcal{G}_{\mathcal{M}}$ with the classical mean-payoff objective from v [7][1]. This value thus equals the mean payoff of a cycle in $\mathcal{G}_{\mathcal{M}}$ and is of the form $\frac{a}{b}$ where $a \in \{0,\ldots,W\cdot|V|\}$ and $b \in \{1,\ldots,|V|\}$. Thus, a binary search is done over $W\cdot|V|^2$ many values, and the two-player game is solved polynomially many times.

The strategy σ^*_ε for BWC synthesis for the BWMP objective is similar to FWMP(ℓ) objective with the difference that the strategy for achieving $\{\varphi_{\mathsf{BWMP}} \geq \lambda_v\}$ from v is memoryless [7].

Probabilistic guarantee. The algorithm for $\mathsf{BPT}((p,0),\beta)$ synthesis for the φ_{BWMP} objective is almost the same as Algorithm 2 with the difference that in Line 3 to compute the maximum μ_M, we use $\{\varphi_{\mathsf{BWMP}} \geq \mu_M\}$ instead of $\{\varphi_{\mathsf{FWMP}(\ell)} \geq \mu_M\}$. The problem of determining if the player almost-surely satisfies the threshold objective $\{\varphi_{\mathsf{BWMP}} \geq \mu_M\}$ from every vertex v in the MDP $\mathcal{M}_0$ restricted to the MEC M is in NP ∩ coNP by reducing it to a polynomial number of calls to two-player classical mean-payoff games [7]. Here also an optimal strategy for $\mathsf{BPT}((p,0),\beta)$ synthesis for the φ_{BWMP} objective may need both memory and randomisation.

Almost-sure guarantee. Again, the algorithm is similar to BAS synthesis for the $\varphi_{\mathsf{FWMP}(\ell)}$ objective. The BAS synthesis problem for the φ_{BWMP} objective is in NP ∩ coNP since computing μ_M for each vertex is in NP ∩ coNP. In contrast to almost-sure satisfaction of $\{\varphi_{\mathsf{FWMP}(\ell)} \geq \mu_M\}$, optimal deterministic memoryless strategies exist for almost-sure satisfaction of $\{\varphi_{\mathsf{BWMP}} \geq \mu_M\}$ [7]. It follows that optimal deterministic memoryless strategies exist for the BAS synthesis problem for the φ_{BWMP} objective.

We thus have the following.

Theorem 5.1. *The* BWC, BPT*, and* BAS *synthesis problems for the* φ_{BWMP} *objective are all in* NP ∩ coNP*, and*

1. *if* $v \models \mathsf{BWC}(\alpha,\beta)$*, then for every* $\varepsilon > 0$*, there exists an* ε*-optimal finite-memory deterministic strategy from* v*.*
2. *if* $v \models \mathsf{BPT}((p,\alpha),\beta)$*, then there exists an optimal finite-memory randomised strategy from* v*.*
3. *if* $v \models \mathsf{BAS}(\alpha,\beta)$*, then there exists an optimal deterministic memoryless strategy from* v*.*

Thus, the complexities achieved are no more than that for sure satisfaction of $\{\varphi_{\mathsf{BWMP}} \geq 0\}$ in a two-player game or expectation maximisation for the φ_{BWMP} objective in an MDP.

6 CONCLUSION

Expectation maximisation with guarantees is a natural problem of importance and interest and appears in various real-world contexts. Further, window mean-payoff objective strengthens classical mean-payoff objective and prevents some undesired behaviours of classical mean payoff. We have shown that the BWC, BAS, and the BPT synthesis of fixed window mean-payoff objectives for MDPs are in PTIME while the problems are in NP ∩ coNP for the bounded window mean-payoff objective. We note that the BWC synthesis problem for classical mean payoff is already in NP ∩ coNP [10, 18]. Our results establish that these problems can be solved at no additional cost than solving the expectation problem for the window mean-payoff objectives while not providing any guarantee, or solving the window mean-payoff objectives with the guarantees while disregarding any requirement on the expected behaviour.

As part of future work, we would like to extend the notion of beyond worst-case and beyond almost-sure to other finitary objectives. It would also be interesting to study these problems in the context of stochastic games.

[1] Two-player games with the $\mathsf{BWMP}(\lambda)$ objective are solved by reducing it to two-player games with total payoff [11].

REFERENCES

[1] S. Almagor, O. Kupferman, and Y. Velner. 2016. Minimizing Expected Cost Under Hard Boolean Constraints, with Applications to Quantitative Synthesis. In *CONCUR (LIPIcs, Vol. 59)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 9:1–9:15.

[2] K.R. Apt and E. Grädel. 2011. *Lectures in Game Theory for Computer Scientists*. Cambridge University Press.

[3] C. Baier and J-P. Katoen. 2008. *Principles of model checking*. MIT Press.

[4] R. Berthon, S. Guha, and J.-F. Raskin. 2020. Mixing Probabilistic and non-Probabilistic Objectives in Markov Decision Processes. In *LICS*. ACM, 195–208.

[5] R. Berthon, J-P. Katoen, and T. Winkler. 2024. Markov Decision Processes with Sure Parity and Multiple Reachability Objectives. In *RP (Lecture Notes in Computer Science, Vol. 15050)*. Springer, 203–220.

[6] R. Berthon, M. Randour, and J.-F. Raskin. 2017. Threshold Constraints with Guarantees for Parity Objectives in Markov Decision Processes. In *ICALP (LIPIcs, Vol. 80)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 121:1–121:15.

[7] B. Bordais, S. Guha, and J.-F. Raskin. 2019. Expected Window Mean-Payoff. In *FSTTCS (LIPIcs, Vol. 150)*. 32:1–32:15.

[8] T. Brázdil, A. Kučera, and P. Novotnỳ. 2016. Optimizing the expected mean payoff in energy Markov decision processes. In *ATVA*. Springer, 32–49.

[9] T. Brihaye, F. Delgrange, Y. Oualhadj, and M. Randour. 2020. Life is Random, Time is Not: Markov Decision Processes with Window Objectives. *Logical Methods in Computer Science* Volume 16, Issue 4 (12 2020).

[10] V. Bruyère, E. Filiot, M. Randour, and J.-F. Raskin. 2017. Meet your expectations with guarantees: Beyond worst-case synthesis in quantitative games. *Information and Computation* 254 (2017), 259–295.

[11] K. Chatterjee, L. Doyen, M. Randour, and J.-F. Raskin. 2015. Looking at mean-payoff and total-payoff through windows. *Information and Computation* 242 (2015), 25–52.

[12] K. Chatterjee, A. Elgyütt, P. Novotný, and O. Rouillé. 2018. Expectation Optimization with Probabilistic Guarantees in POMDPs with Discounted-Sum Objectives. In *IJCAI*. ijcai.org, 4692–4699.

[13] K. Chatterjee and M. Henzinger. 2014. Efficient and Dynamic Algorithms for Alternating Büchi Games and Maximal End-Component Decomposition. *J. ACM* 61, 3 (2014), 15:1–15:40. https://doi.org/10.1145/2597631

[14] K. Chatterjee, T. A. Henzinger, and F. Horn. 2009. Stochastic Games with Finitary Objectives. In *MFCS*. Springer Berlin Heidelberg, 34–54.

[15] K. Chatterjee, Z. Křetínská, and J. Křetínský. 2017. Unifying Two Views on Multiple Mean-Payoff Objectives in Markov Decision Processes. *Logical Methods in Computer Science* Volume 13, Issue 2 (July 2017).

[16] K. Chatterjee, P. Novotný, G. A. Pérez, J.-F. Raskin, and D. Zikelic. 2017. Optimizing Expectation with Guarantees in POMDPs. In *AAAI*. AAAI Press, 3725–3732.

[17] K. Chatterjee and N. Piterman. 2019. Combinations of Qualitative Winning for Stochastic Parity Games. In *CONCUR (LIPIcs, Vol. 140)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 6:1–6:17.

[18] L. Clemente and J.-F. Raskin. 2015. Multidimensional beyond worst-case and almost-sure problems for mean-payoff objectives. In *LICS*. IEEE, 257–268.

[19] L. Doyen, P. Gaba, and S. Guha. 2024. Stochastic Window Mean-payoff Games. In *FoSSaCS Part I (LNCS, Vol. 14574)*. Springer, 34–54.

[20] A. Ehrenfeucht and J. Mycielski. 1979. Positional Strategies for Mean Payoff Games. *Int. Journal of Game Theory* 8, 2 (1979), 109–113.

[21] P. Gaba and S. Guha. 2025. Optimising expectation with guarantees for window mean payoff in Markov decision processes. *CoRR* abs/2501.05384 (2025).

[22] G. Geeraerts, S. Guha, and J.-F. Raskin. 2018. Safe and Optimal Scheduling for Hard and Soft Tasks. In *FSTTCS*. 36:1–36:22.

[23] E. Grädel, W. Thomas, and T. Wilke (Eds.). 2002. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*. Lecture Notes in Computer Science, Vol. 2500. Springer.

[24] L.G. Khachiyan. 1980. Polynomial algorithms in linear programming. *U. S. S. R. Comput. Math. and Math. Phys.* 20, 1 (1980), 53–72.

[25] M.L. Puterman. 1994. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley and Sons.

[26] M. Y. Vardi. 1985. Automatic Verification of Probabilistic Concurrent Finite-State Programs. In *FOCS*. IEEE Computer Society, 327–338.

[27] U. Zwick and M. Paterson. 1996. The Complexity of Mean Payoff Games on Graphs. *Theor. Comput. Sci.* 158, 1&2 (1996), 343–359.