Expected Window Mean-Payoff

Benjamin Bordais[1], Shibashis Guha[2], and Jean-François Raskin[2]

[1] ENS Rennes
[2] Université Libre de Bruxelles

Abstract. In the window mean-payoff objective, given an infinite path, instead of considering a long run average, we consider the mean-payoff over a finite length window that slides over the path. In fact, we consider a function that, given a path, returns the supremum value of the mean-payoff that can be ensured over all windows either from the very beginning of the path – that is the prefix dependent version – or from some position on – that is the prefix independent version. Then, we compute the expected value of that function in the context of (weighted) Markov decision process (MDP in short) and in the special case of Markov chain (MC in short). In the case of the prefix independent version, we show that the problem of computing the expected value in MDPs can be done by solving a two-player game, and that this problem is at least as hard as solving that two-player game. When a specific window length is given, we have pseudo-polynomial algorithm (polynomial in the window size that is given in binary), when we consider the supremum over all possible lengths, the algorithm we have is in NP $\cap$ coNP. As for the prefix dependent version we have an algorithm that is exponential in the window length l to compute the expected value in MDPs, but we also have a PP-hardness result for that problem, even when l is given in unary. We also provide algorithms for the special case of MCs.

1 Introduction

Markov Decision Processes (MDPs) are a classical model for decision-making inside stochastic environments [14,1]. In that context, a stochastic model of the environment is formalized and we aim at finding strategies that maximize the expected performance of the system with that stochastic environment. This performance in turn is formalized by a function that maps each infinite path in the MDP to a value. One classical such function is the mean-payoff function that maps an infinite path to the limit of the means of the payoffs obtained on its prefixes. While this measure is classical, alternatives to the mean-payoff measure have been studied in the literature, e.g. one of the most studied alternative notion is the notion of discounted sum [14]. The main drawback of the mean-payoff value is that it does not guarantee local stability of the values along the path: if the limit mean-value of an infinite path converges towards a value v, it may be the case that for arbitrarily long infixes of the infinite path, the mean payoff of the infix is largely away from v. There have been several recent contributions [7,4,8,5] in the literature to deal with these possible fluctuations from the

mean-payoff value along a path. In this paper, we study the notion of window mean-payoff that was introduced in [7,8] for two-player games in the context of MDPs, and we provide algorithms and prove computational complexities for the expected value of window mean-payoff objectives.

As introduced in [8], in a window mean-payoff objective instead of the limit of the mean-payoffs along the whole sequence of payoffs, we consider payoffs over a local finite length window sliding along the infinite sequence: the objective asks that the mean-payoff must always reach a given threshold within the window length l. This objective is clearly a strengthening of the mean-payoff objective: for all length l, all infinite sequences π of payoffs that satisfy the window mean-payoff objective for threshold λ implies that π has a mean-payoff value larger than or equal to λ.

In this paper, we study how to maximize the expected value of the window mean-payoff function f^l_{WMP} in MDPs. The value of an infinite sequence of integer values $\pi : \mathbb{N} \to \mathbb{Q}$ for this function is defined as follows:

$$f^l_{WMP}(\pi) = \sup\{\lambda \in \mathbb{R} \mid \forall i \geq 0 : \max_{1 \leq j \leq l} \frac{1}{j} \sum_{m=0}^{j-1} \pi(i+m) \geq \lambda\}$$

i.e., it returns the supremum of all window mean-payoff thresholds that are enforced by the sequence of payoffs π. As in [12], we study natural variants of this measure: (i) when the size of the window is fixed or when it is left unspecified but needs to be finite, and (ii) when the window property needs to be true from the beginning of the path, or a prefix independent version which asks the window property to eventually hold from some position in the path.

Main contributions Our results are as follows. First, for the prefix independent version of the measure f^l_{WMP} and for a fixed window length l, we provide an algorithm to compute the best expected value of f^l_{WMP} with a time complexity that is polynomial in the size of the MDP Γ and in l (Theorem 1). It is worth to note that, since the main motivation for introducing the window mean-payoff objective is to ensure strong stability over reasonable period of time, it is very natural to assume that l is bounded polynomially by the size of the MDP Γ. This in turn implies that our algorithm is fully polynomial for the most interesting cases. We also note that this complexity matches the complexity of computing the value of the function f^l_{WMP} for two-player games [8], and we provide a relative hardness result: the problem of deciding the existence of a winning strategy in a two-player window mean-payoff game can be reduced to the problem of deciding if the maximal expected mean-payoff value of a MDP for f^l_{WMP} is larger than or equal to a given threshold λ (Theorem 2).

Second, we consider the case where the length l in the measure f^l_{WMP} is not fixed but only required to be finite. In that case, we provide an algorithm which is in NP $\cap$ coNP (Theorem 4). In addition, we show that providing a polynomial time solution to our problem would also provide a polynomial time solution to the value problem in mean-payoff games (Theorem 5), this is a long-standing open problem in the area [17].

Third, we consider the case where the good window property needs to be imposed from the start of the path (for a fixed length). In that case, surprisingly, the problem of computing if there is a strategy to obtain an expected value above a threshold λ is harder than for two-player games unless P=PP. Indeed, while the threshold problem for the worst-case value can be solved in time polynomial in the size of the game and in l, we show that for the expected value in an MDP, the problem is PP-Hard even if l is given in unary (Theorem 8). To solve the problem, we provide an algorithm that executes in time which is polynomial in the size of the MDP, polynomial in the largest payoff appearing in the MDP, and exponential in the length l (Theorem 7).

Finally, while our main results concentrate on MDPs, we also systematically provide results for the special case of MCs.

Related Works As already mentioned, the window mean-payoff objective was introduced in [7] for two-player games. We show in this paper that the complexity of computing maximal expected value for the window mean-payoff function is closely related to the computation of the worst-case value of a game inside the end-components of the MDP (see Lemma 1 and 3) for the prefix independent version of our objective. The window mean-payoff objectives were also considered in games with imperfect information in [12], and in combination with omega-regular constraints in [6].

Stability issues of the mean-payoff measure have been studied in several contributions. In [4], the authors study MDP where the objective is to optimize the expected mean-payoff performance and stability. They propose alternative definitions to the classical notions of statistical variance. The notion of stability offered by window mean-payoff objective and studied in this paper is stronger than one proposed in that paper. The techniques needed to solve the two problems are very different too as they mainly rely on solving sets of quadratic constraints.

In [5], window-stability objectives have been introduced. Those objective are inspired from the window mean-payoff objective of [4] but they are different in that they do not enjoy the so called inductive window property because of the stricter stability constraints that those objectives impose. The authors have considered the window-stability objectives in the context of games (2 players) and graphs (1 player) but they did not consider the case of MDPs ($1\frac{1}{2}$ players).

MDP with classical mean-payoff objectives have been extensively studied both for the probabilistic threshold and the expectation payoff problem, see e.g. [14]. Combination of both type of constraints have been considered in [3].

Due to lack of space, we only provide sketches of the proofs in this paper. A complete version of this work with full proofs appears in [2].

2 Preliminaries

For $k \in \mathbb{N}$, we denote by $[k]_0$ and $[k]$ the set of natural numbers $\{0, \ldots, k\}$ and $\{1, \ldots, k\}$ respectively. Given a finite set A, a (rational) *probability distribution*

over A is a function $P\colon A \to [0,1] \cap \mathbb{Q}$ such that $\sum_{a \in A} P(a) = 1$. We denote the set of probability distributions on A by $\mathcal{D}(A)$.

2.1 Weighted Markov decision processes and Markov chains

Markov chains A finite weighted *Markov chain* (MC, for short) is a tuple $\mathcal{M} = \langle S, E, s_{init}, w, \mathbb{P} \rangle$, where S is the finite set of states, $s_{init} \in S$ is the initial state of this Markov chain, $E \subseteq S \times S$ is the set of edges, the function $w : E \mapsto \mathbb{Q}$ defines the *weight*s (or *payoff*s) of the edges, and $\mathbb{P} : S \to \mathcal{D}(E)$ is a function that assigns a probability distribution – on the set $E(s)$ of outgoing edges from s – to all states $s \in S$. In the following, $\mathbb{P}(s,(s,s'))$ is denoted $\mathbb{P}(s,s')$, for all $s, s' \in S$. The size of $\mathcal{M}$ is the number of states $|S|$, and will be denoted $|\mathcal{M}|$.

For a state $s \in S$, we define the set of infinite paths in $\mathcal{M}$ starting from s as $Paths^{\mathcal{M}}(s) = \{\pi = s_0 s_1 \ldots \in S^\omega \mid s_0 = s, \forall\, n \in \mathbb{N}, \mathbb{P}(s_n, s_{n+1}) > 0\}$. The set of all the paths in $\mathcal{M}$ is $Paths^{\mathcal{M}} = \bigcup_{s \in S} Paths^{\mathcal{M}}(s)$. For a path $\pi = s_0 \ldots \in Paths^{\mathcal{M}}$, by $\pi(i,l)$ we denote the sequence of $l+1$ states (or l edges) $s_i \ldots s_{i+l}$. The infinite path of π starting in s_n is denoted $\pi(n,\infty) \in Paths^{\mathcal{M}}$.

Consider some measurable function $f : Paths^{\mathcal{M}}(s_{init}) \to \mathbb{R}$ associating a value to each infinite path starting from s_{init}. For an interval $I \subset \mathbb{R}$, we denote by $f^{-1}(\mathcal{M}, s_{init}, I)$ the set $\{\pi \in Paths^{\mathcal{M}}(s_{init}) \mid f(\pi) \in I\}$, and for $r \in \mathbb{R}$, $f^{-1}(\mathcal{M}, s_{init}, r)$ refers to $f^{-1}(\mathcal{M}, s_{init}, [r,r])$. Since the set of paths $Paths^{\mathcal{M}}(s_{init})$ forms a probability space, measured by a function Pr, and f is a random variable, we denote by $\mathbb{E}^{\mathcal{M}}_{s_{init}}(f) = \int_{x \in \mathbb{R}} Pr(f^{-1}(\mathcal{M}, s_{init}, x)) \cdot x$ the *expected value* of f over the set of paths starting from s_{init}.

The bottom strongly connected components (BSCCs for short) in a Markov chain $\mathcal{M}$ are the strongly connected components from which it is impossible to exit (for all $s \in \mathcal{B}$ and $t \in \mathcal{M}$, we have $\mathbb{P}(s,t) > 0$ implies that $t \in \mathcal{B}$). We denote by $BSCC(\mathcal{M})$ the set of BSCCs of the Markov chain $\mathcal{M}$. Every infinite path eventually ends up in one of the BSCCs almost surely. Formally:

Proposition 1. *For all state $s \in S$, we have: $Pr(\pi \in Paths^{\mathcal{M}}(s) \mid \exists \mathcal{B} \in BSCC(\mathcal{M}),\ \pi \models \Diamond\Box\mathcal{B}) = 1$.*

Markov decision process A finite weighted *Markov decision process* (MDP, for short) is a tuple $\Gamma = \langle S, E, Act, s_{init}, w, \mathbb{P} \rangle$, where S is the set of states, $s_{init} \in S$ is the initial state of this Markov decision process, Act is the set of actions, and $E \subseteq S \times Act \times S$ is set of edges. The function $w : E \mapsto \mathbb{Q}$ defines the *weight*s of the edges, and $\mathbb{P} : S \times Act \to \mathcal{D}(E)$ is a function that assigns a probability distribution – on the set $E(s,a)$ of outgoing edges from s – to all states $s \in S$ if action $a \in Act$ is taken in s. Given $s \in S$ and $a \in Act$, we define $Post(s,a) = \{s' \in S \mid \mathbb{P}(s,a,s') > 0\}$. Then, for all state $s \in S$, we denote by $Act(s)$ the set of actions $\{a \in Act \mid Post(s,a) \neq \emptyset\}$. We assume that, for all $s \in S$, $Act(s) \neq \emptyset$. The size of Γ will be denoted $|\Gamma|$, and will refer to the number of states of Γ times the number of actions, that is $|S| \cdot |Act|$.

A strategy in Γ is a function $\sigma : S^+ \mapsto Act$ such that $\sigma(s_0 \ldots s_n) \in Act(s_n)$, for all $s_0 \ldots s_n \in S^+$. We denote by $\mathsf{strat}(G)$ the set of strategies available in Γ.

Once we fix a strategy σ in an MDP $\Gamma = \langle S, E, Act, s_{init}, w, \mathbb{P}\rangle$, we obtain an MC $\Gamma^{[\sigma]}$. Consider a measurable function f that associates a value to infinite paths in Markov chains. Then, the expected value of f in an MDP Γ, that is $\mathbb{E}^{\Gamma}_{s_{init}}(f)$ is equal to $\sup_{\sigma \in \mathsf{strat}(\Gamma)} \mathbb{E}^{\Gamma^{[\sigma]}}_{s_{init}}(f)$.

An end component (EC for short) $M = (T, A) \subseteq S \times Act$ is a sub-MDP of Γ (that is, that ensures that, for all $s \in T, a \in Act(s) \cap A$, we have $Post(s, a) \subseteq T$) that is strongly connected. A maximal EC (MEC for short) is an EC that is not included in any other EC. We denote by $MEC(\Gamma)$ the set of all maximal end components of Γ. Any infinite path will eventually end up in one maximal end component almost surely, whatever strategy is considered. That is stated in the following proposition:

Proposition 2. *For all strategy $\sigma \in \mathsf{strat}(\Gamma)$, for all state $s \in T$, we have: $Pr(\pi \in Paths^{\Gamma^{[\sigma]}}(s) \mid \exists M \in MEC(\Gamma),\ \pi \models \Diamond\Box M) = 1$.*

2.2 Weighted Two-Player Games

We consider weighted two-player games $G = \langle S_1, S_2, s_{init}, E, w\rangle$ where the set of vertices $S = S_1 \uplus S_2$ is partitioned into the vertices belonging to Player 1, that is S_1, and the vertices belonging to Player 2, that is S_2, and $s_{init} \in S_1$ is the initial vertex. The set of edges $E \subseteq S_1 \times S_2 \cup S_2 \times S_1$ is such that for all $s \in S$, there exists $s' \in S$ such that $(s, s') \in E$. For all $s \in S$, we denote by $Succ(s) = \{s' \in S \mid (s, s') \in E\}$. The weight function[3] w is such that $w : E \cap S_2 \times S_1 \to \mathbb{Q}$. An MDP $\Gamma = \langle S, E, Act, s_{init}, w, \mathbb{P}\rangle$ can be transformed into a two-player game $G_\Gamma = \langle S_1, S_2, s_{init}, E, w'\rangle$ where $S_1 = S$, $S_2 = \{(t, a) \in S \times Act \mid a \in Act(t)\}$, $E = E_1 \cup E_2$ with $E_1 = \{(t, (t, a)) \in S_1 \times S_2\}$ and $E_2 = \{((t, a), t') \in S_2 \times S_1 \mid t' \in Post(t, a)\}$. Further, $w' : E_2 \mapsto \mathbb{Q}$ with $w'((t, a), t') = w(t, a, t')$.

The strategies available for the two players are defined in the same way that we defined strategies in MDPs. The set of strategies for Player 1 and Player 2 are denoted $\mathsf{strat}_1(G)$ and $\mathsf{strat}_2(G)$ respectively.

In the following, we will denote the size of G by $|G|$ the number of states of G, that is $|S_1 \uplus S_2|$.

[3] We do not consider weight on the edges coming from states belonging to Player 1 so that a two-player game can be seen as an MDP where the actions are states belonging to Player 2, who is not a stochastic adversary (that is, Player 2 uses deterministic strategy, analogous to the strategy defined in MDPs). That definition of two-player games is not exactly the one used in the previous paper dealing with window mean-payoff (that is [7]). However, their definition of games (where the weight function is defined on every edges, not only the edges chosen by Player 2) can be easily translated into the one we use by doubling the number edges. That does not affect the asymptotic complexity of the algorithms considered.

3 Window mean-payoff

Let $\mathcal{M} = \langle S, E, s_{init}, w, \mathbb{P} \rangle$ be a finite Markov chain. Consider a window length $l_{max} \geq 1$ and a sequence of edges $\rho = e_1 \dots e_{l_{max}}$ in $\mathcal{M}$. We define the window mean-payoff of ρ, that is $WMP(\rho)$, by $WMP(\rho) = \max_{k \in [l_{max}]} \frac{1}{k} \sum_{i=1}^{k} w(e_i)$.

The value $WMP(\rho)$ is the maximum mean-payoff one can ensure over a window of length $k \in [l_{max}]$. For a given infinite path $\pi = s_0 \dots$, a threshold $\lambda \in \mathbb{Q}$, a position $i \in \mathbb{N}$ and $l \in [l_{max}]$, we say that the window s_i is *closed* in s_{i+l} with respect to λ if $WMP(\pi(i,l)) \geq \lambda$. Otherwise, the window is *open*.

We define the *fixed window mean-payoff function* $f^{l_{\max}}_{FixWMP} : Paths^{\mathcal{M}} \mapsto \mathbb{R}$ such that, for every path $\pi = s_0 \dots \in Paths^{\mathcal{M}}$:

$$f^{l_{\max}}_{FixWMP}(\pi) = \sup\{\lambda \in \mathbb{R} \mid \exists k \in \mathbb{N},\ \forall i \geq k : WMP(\pi(i, l_{max})) \geq \lambda\} \qquad (1)$$

The value $f^{l_{\max}}_{FixWMP}(\pi)$ corresponds to the supremum over all threshold λ that are above every window mean-payoff for length l_{max} from some position on. This function is prefix independent, that is, for every path $\pi \in Paths^{\mathcal{M}}$, for all $n \geq 1$, $f^{l_{\max}}_{FixWMP}(\pi) = f^{l_{\max}}_{FixWMP}(\pi(n, \infty))$.

Then, we define the *bounded window mean-payoff function* $f_{BWMP} : Paths^{\mathcal{M}} \mapsto \mathbb{R}$ such that, for every path $\pi = s_0 \dots \in Paths^{\mathcal{M}}$:

$$f_{BWMP}(\pi) = \sup\{\lambda \in \mathbb{R} \mid \exists l, k \geq 1, \forall i \geq k : WMP(\pi(i,l)) \geq \lambda\} \qquad (2)$$

The value $f_{BWMP}(\pi)$ corresponds to the supremum over all threshold λ that ensures that there is a length l for which every window mean-payoff for that length l are above λ from some position on. That function is also prefix independent.

Finally, we define the *direct fixed window mean-payoff function* $f^{l_{\max}}_{DirFixWMP} : Paths^{\mathcal{M}} \mapsto \mathbb{R}$ such that, for every path $\pi = s_0 \dots \in Paths^{\mathcal{M}}$:

$$f^{l_{\max}}_{DirFixWMP}(\pi) = \sup\{\lambda \in \mathbb{R} \mid \forall i \geq 0 : WMP(\pi(i, l_{max})) \geq \lambda\} \qquad (3)$$

The value $f^{l_{\max}}_{DirFixWMP}(\pi)$ corresponds to the supremum over all threshold λ that are above every window mean-payoff for length l_{max} from the very beginning of the path. That function is not prefix independent. Note that for any path $\pi \in Paths^{\mathcal{M}}$, $f^{l_{\max}}_{DirFixWMP}(\pi) \leq f^{l_{\max}}_{FixWMP}(\pi)$.

We also define the *mean-payoff* function $f_{Mean} : Paths^{\mathcal{M}} \mapsto \mathbb{R}$ such that, for $\pi = s_0 \dots \in Paths^{\mathcal{M}}$, we have $f_{Mean}(\pi) = \liminf_{n \to \infty} \frac{1}{n} \sum_{k=0}^{n-1} w(s_k, s_{k+1})$.

In the following, in the two-player games, MCs and MDPs w.l.o.g. we consider only non-negative integer weights[4].

[4] Note that if the weights belong to $\mathbb{Q}$, then one can multiply them with the LCM d of their denominators to obtain integer weights. Among the resultant set of integer weights, if the minimum integer weight κ is negative, then we add -κ to the weight of each edge so as to obtain weights that are natural numbers. In that case, if the expected value of some function defined above is equal to x before the change, then the new expected value is equal to $d \cdot x - \kappa$.

4 Fixed Window Mean-Payoff

Expected value in an MDP We first consider the expected value of the fixed window mean-payoff function for some length l_{max}, that is $f^{l_{max}}_{FixWMP}$, in an MDP $\Gamma = \langle S, E, Act, s_{init}, w, \mathbb{P}\rangle$.

Recall that, by Proposition 2, we have that for every strategy σ, each path $\pi \in Paths^{\Gamma^{[\sigma]}}$ almost surely ends up in an MEC. Since $f^{l_{max}}_{FixWMP}$ is prefix independent, the value of a path only depends on the MEC in which it ends up. Now, consider an MEC $M = (T, A) \in MEC(\Gamma)$ of Γ and a state $s \in T$. Since M is strongly connected (since it is an MEC), for every pair of states $s, s' \in T$, there exists a reaching strategy $\sigma_{(s,s')} \in \mathsf{strat}(M)$ such that every path starting from s reaches s' almost surely, in the Markov chain $M^{[\sigma_{(s,s')}]}$. Therefore, for all $s, s' \in T$, we have $\mathbb{E}^{\Gamma}_s(f^{l_{\max}}_{FixWMP}) = \mathbb{E}^{\Gamma}_{s'}(f^{l_{\max}}_{FixWMP})$. Let us denote by $\lambda^{l_{max}}_M$ the expected value of $f^{l_{\max}}_{FixWMP}$ among all the paths that and up in M. In that case, we have the following lemma:

Lemma 1. *Let $M = (T, A)$ be an MEC of Γ. Then we have:*

$$\lambda^{l_{max}}_M = \max_{s\in T} \underbrace{\sup_{\sigma_1 \in \mathsf{strat}_1(G_M)} \inf_{\sigma_2 \in \mathsf{strat}_2(G_M)} f^{l_{\max}}_{DirFixWMP}(\pi_{(G_M,s,\sigma_1,\sigma_2)})}_{denoted\ g(s)}$$

Proof (sketch). Let $v \in T$ be a state that maximizes the outcome of the two-player game (that is, that ensures that $g(v) = \max_{s\in T} g(s)$). Then, once a strategy σ in the MDP is fixed, because we consider infinite paths in the induced MC $\Gamma^{[\sigma]}$, every possible (with respect to the strategy σ) finite sequence of states will be visited infinitely often almost surely. In particular, the worst sequence of transitions in terms of maximizing the fixed window mean-payoff (that is the sequence that Player 2 chooses in the two-player game G_M) is visited infinitely often almost surely. Therefore, what happens in the MEC M is analogous to what happens in the two-player game G_M. Then, the maximum over all states of the outcome of the two-player game for the fixed or direct fixed window mean-payoff function are identical. The lemma follows. □

Once the expected value inside every MEC is computed, we construct a new MDP Γ^{MEC} that is equal to Γ except that we replace the weight of the edges in each MEC M by $\lambda^{l_{max}}_M$. Then we run an expected mean-payoff algorithm on Γ^{MEC} and the value obtained is equal to the expected window mean-payoff over the whole MDP Γ. Further, the expected mean-payoff in Γ^{MEC} can be computed in time that is polynomial in the size of the MDP Γ^{MEC}. Therefore, we have the following theorem (since $|\Gamma^{MEC}| = |\Gamma|$):

Theorem 1. *Computing the expected value of $f^{l_{\max}}_{FixWMP}$ in an MDP Γ can be done in time $O(p_1(|\Gamma|, l_{\max}))$ where p_1 is a polynomial function.*

Since $l_{\max}$ is given in binary, the complexity we have is in fact exponential in the binary length encoding of l_{max}.

Note that the best algorithm we know (see [7]) to solve a two-player game G for the direct fixed window mean-payoff objective for length $l_{\max}$ runs in time $O(p_2(|G|, l_{\max}))$ where p_2 is also a polynomial function. In fact, we can show that solving the two-player game for the direct fixed window mean-payoff objective can be reduced in polynomial time to computing the expected value of the fixed window mean-payoff function. Formally:

Theorem 2. *Computing the expected value of the fixed window mean-payoff function in an MDP is at least as hard as solving a two-player game for the direct fixed window mean-payoff objective (for polynomial reductions).*

Proof (sketch). Consider a two-player game $G = \langle S_1, S_2, s_{init}, E, w\rangle$ that we want to solve for the direct fixed window mean-payoff objective. We first modify the game G into another game G^{reset} by adding, from every state in S_2, an edge to s_{init} with a very high weight (for instance, $(W+1)\cdot l_{max}$, where W is the maximum weight appearing in G). Now, the game G^{reset} is strongly connected. Moreover, if the weight on the new edges is high enough, such a new edge will not be interesting for Player 2 to take infinitely often. However, it may be taken finitely many times, by Player 2 if it is interesting to reach the state s_{init}. In that way, the maximum outcome that be achieved over all starting state is done in state s_{init} and the outcome of the game G^{reset} from s_{init} is the same as the outcome of the game G from s_{init}. Then, we conclude by using Lemma 1. □

Expected value in an MC Consider a Markov chain $\mathcal{M} = \langle S, E, s_{init}, w, \mathbb{P}\rangle$. The techniques used here are very similar to the ones used in MDPs. By Proposition 1, each path will almost surely end up in a BSCC. Therefore, we first consider a BSCC $\mathcal{B} \in BSCC(\mathcal{M})$. By definition, $\mathcal{B}$ is strongly connected. Therefore, the expected value of $f^{l_{max}}_{FixWMP}$ (that is prefix-independent) is the same from every state. Let us denote by $\mu^{l_{max}}_{\mathcal{B}}$ the expected value of $f^{l_{\max}}_{FixWMP}$ over all paths that are in $\mathcal{B}$. Then:

Lemma 2. *Let $\mathcal{B} \in BSCC(\mathcal{M})$. Then:*

$$\mu^{l_{max}}_{\mathcal{B}} = \underbrace{\min_{s\in\mathcal{B}} \min_{\pi\in Paths^{\mathcal{B}}} WMP(\pi(0, l_{max}))}_{denoted\ m_{\mathcal{B}}}$$

Proof (sketch). Because $\mathcal{B}$ is strongly connected, every finite sequence of states in $\mathcal{B}$ is visited infinitely often almost surely. In particular, the sequence that minimizes the window mean-payoff in $\mathcal{B}$ (whose window mean-payoff is equal to $m_{\mathcal{B}}$) is seen infinitely often. Hence the lemma. □

Note that this also corresponds to the outcome of the game where every state belongs to Player 2 for the direct fixed window mean-payoff function. Therefore, by using the same algorithm that computed $\lambda^{l_{max}}_M$ in an MEC M, $\mu^{l_{max}}_{\mathcal{B}}$ can be computed in time polynomial in l_{max}

The set of BSCCs and the probability of reaching each BSCC can be computed in polynomial time. Moreover, we have the following equality: $\mathbb{E}^{\mathcal{M}}_{s_{init}}(f^{l_{max}}_{FixWMP}) = \sum_{\mathcal{B}\in BSCC(\mathcal{M})} Pr(Reach_{s_{init}}(\mathcal{B}))\cdot m_{\mathcal{B}}$.

Hence, the following theorem:

Theorem 3. *Finding the expected value of $f_{FixWMP}^{l_{\max}}$ in an MC $\mathcal{M}$ can be done in time $O(q_1(|\mathcal{M}|, l_{\max}))$ where q_1 is a polynomial function.*

We did not find a truly polynomial algorithm to compute $\lambda_M^{l_{max}}$ in an MEC M. Hence the complexity. It is still an open problem to know if there exists a truly polynomial algorithm that computes $\mu_{\mathcal{B}}^{l_{max}}$.

5 Bounded Window Mean-Payoff

Expected value in MDPs We are interested in the expected value of the bounded window mean-payoff function f_{BWMP} in an MDP $\Gamma = \langle S, E, Act, s_{init}, w, \mathbb{P}\rangle$. As in the case of the fixed window mean-payoff function, the bounded window mean-payoff function too is prefix independent. Therefore, we will use techniques very similar to the one we used in Section 4.

For an MEC $M = (T, A) \in MEC(\Gamma)$, we denote by λ_M the expected value of the bounded window mean-payoff considering the paths that end up in M. Recall that f_{Mean} is the mean-payoff function. Then:

Lemma 3. *Let $M = (T, A)$ be an MEC of Γ. Then we have:*

$$\lambda_M = \max_{s \in T} \underbrace{\sup_{\sigma_1 \in \mathsf{strat}_1(G_M)} \inf_{\sigma_2 \in \mathsf{strat}_2(G_M)} f_{Mean}(\pi_{(G_M, s, \sigma_1, \sigma_2)})}_{\text{denoted } \bar{g}(s)}$$

Proof (sketch). Let s be the state that gives the maximum of the mean-payoff value over all states. Since the bounded window mean-payoff is the supremum of the window mean-payoff over all possible window lengths and there exists a strategy such that almost-surely in M state s can be reached from the other states, the result follows. □

Solving a two-player game with the mean-payoff objective is known to be in NP ∩ coNP and the existence of a polynomial algorithm is an open question [17]. Once the expected value inside every MEC M is computed, we use the same method as in the fixed window mean-payoff case (we construct a new MDP in which we compute the expected mean-payoff), which requires a polynomial time algorithm. Hence we have the following theorem:

Theorem 4. *Deciding whether or not the expected value of f_{BWMP} in an MDP is above some threshold λ is in* NP ∩ coNP.

Then, with the same reduction used to prove Theorem 2 and with Lemma 3, we can show the following:

Theorem 5. *Computing the expected value of the bounded window mean-payoff in an MDP is at least as hard as solving a two-player game for the mean-payoff objective.*

Expected value in MCs Consider a Markov chain $\mathcal{M} = \langle S, E, s_{init}, w, \mathbb{P} \rangle$. We proceed in the same way as we did for the expected value of the fixed window mean-payoff function, since the function f_{BWMP} is prefix independent. Let $\mathcal{B} \in BSCC(\mathcal{M})$ be a BSCC and let $\mu_{\mathcal{B}}$ be the expected value of f_{BWMP} among all the paths that are in $\mathcal{B}$. We have the following lemma:

Lemma 4. *Let $\mathcal{B} \in BSCC(\mathcal{M})$. Then:*

$$\mu_{\mathcal{B}} = \underbrace{\min_{\rho = s_0 \dots s_k \in ElemCycle(\mathcal{B})} \underbrace{\frac{1}{k} \sum_{i=0}^{k-1} w(s_i, s_{i+1})}_{denoted\ MP(\rho)}}_{denoted\ c_{\mathcal{B}}}$$

where $ElemCycle(\mathcal{B})$ denotes the (finite) set of simple cycles in $\mathcal{B}$.

For any cycle $\rho \in ElemCycle(\mathcal{B})$, the value $MP(\rho)$ corresponds to the mean-payoff of the cycle ρ. The value $c_{\mathcal{B}}$ is the mean of a minimum mean cycle.

Proof (sketch). For every $\epsilon > 0$, we can ensure $c_{\mathcal{B}} - \epsilon$ by considering an appropriate window length. Since f_{BWMP} considers supremum over all $c_{\mathcal{B}} - \epsilon$, the bounded window mean-payoff cannot be below $c_{\mathcal{B}}$. Moreover, for every $n \geq 1$, the sequence of states that cycles n times around the minimum mean cycle ρ is seen infinitely often almost surely, and its window mean-payoff is at most $MP(\rho)$ (if we start in the right point). In fact, for every length $l \geq 1$, the fixed window mean-payoff for length l is almost surely at most $c_{\mathcal{B}}$. Hence, almost surely, the bounded window mean-payoff of a path is at most $c_{\mathcal{B}}$. □

By using the Karp Algorithm (see [13]), it is possible to compute the minimum mean cycle in time polynomial in the size of the BSCC $\mathcal{B}$. Hence, we have the following theorem:

Theorem 6. *Finding the expected value of f_{BWMP} in an MC $\mathcal{M}$ can be done in time $O(q_2(|\mathcal{M}|))$ where q_2 is a polynomial function.*

6 Direct Fixed Window Mean-Payoff

Expected value in MDPs Consider now the function $f^{l_{\max}}_{DirFixWMP}$. We want to compute its expected value in an MDP Γ. Since it is prefix-dependent, it is no longer enough to consider the expected value only inside the MECs.

The algorithm we have consists in constructing a new MDP $\Gamma_{l_{max}}$, and then computing the expected value of f_{Mean} in $\Gamma_{l_{max}}$. Let us denote by S the set of states of the MDP Γ and let W be the maximum weight that appears in Γ. Then, the set of states of $\Gamma_{l_{max}}$, that is S', is equal to $S' = S \times ([W]_0)^{l_{max}-1} \times \Lambda$ where we have $\Lambda = \{\frac{p}{q} \mid q \in [l_{max}],\ p \in [q \cdot W]_0\}$. Informally, the idea of this construction is the following: Consider a state $t = (s, [w_1, \dots, w_{l_{\max}-1}], \lambda_t) \in S'$. This state corresponds to a finite path $\rho = s_0 \dots s$ in Γ. Moreover, the last

$l_{max}-1$ weights encountered in ρ are $w_1,\dots,w_{l_{\max}-1}$. Finally, λ_t keeps track of the minimum window mean-payoff seen so far in ρ. Moreover, the MDP $\Gamma_{l_{max}}$ is constructed in a way so that every edge exiting t has a weight equal to λ_t. In this way, for $\pi \in Paths^{\Gamma_{l_{max}}}$, the sequence of weights seen in π is a non-increasing series included in the finite set Λ. Therefore, eventually that series reaches a fixed point that is the direct fixed window mean-payoff λ of the corresponding path in Γ. Since the series reaches the fixed point λ, then the mean-payoff of that series is equal to λ. In fact, we have the following lemma:

Lemma 5. *For an MDP Γ, we have:*

$$\mathbb{E}^{\Gamma}_{s_{init}}(f^{l_{max}}_{DirFixWMP}) = \mathbb{E}^{\Gamma_{l_{max}}}_{s'_{init}}(f_{Mean})$$

where $s'_{init} = (s_{init}, [W,\dots,W], W)$.

We have $|\Gamma_{l_{\max}}| \leq |S| \cdot W^{l_{\max}} \cdot l^2_{\max}$. Since computing the expected value of the mean-payoff in an MDP can be done in polynomial time (see [14]), we have the following result:

Theorem 7. *Computing the expected value of $f^{l_{\max}}_{DirFixWMP}$ in an MDP Γ can be done in time $O(p_3(|S| \cdot W^{l_{\max}} \cdot l^2_{\max}))$ where p_3 is a polynomial function and W is the maximum weight appearing in the MDP Γ.*

Although the algorithm we have is exponential in $l_{\max}$, and therefore doubly exponential in the binary encoding $l_{\max}$, it has to be noted that it is fixed parameter tractable, if we consider W and $l_{\max}$ to be parameters.

We now consider the hardness of the problem. We show that given an MDP Γ with an initial state s_{init}, a length $l_{\max}$ and a threshold λ, checking if $\mathbb{E}^{\Gamma}_{s_{init}}(f^{l_{max}}_{DirFixWMP}) > \lambda$ is PP-*hard*. Recall that PP is the class of languages $L \subseteq \Sigma^*$ recognized by a probabilistic polynomial-time Turing machine M with access to a fair coin such that for all $w \in \Sigma^*$, we have $w \in L$ if and only if M accepts w with a probability above $\frac{1}{2}$. The class PP contains NP, is closed under complementation [15] and hence also contains the class coNP. Further, the class PP is contained in PSPACE.

Theorem 8. *The direct fixed window mean-payoff problem for MDP is PP-hard.*

We show a reduction from k-th largest subset which has recently been shown to be PP-complete [11]. The k-th largest subset problem is stated as given a finite set of positive integer $A = \{a_1,\dots,a_n\}$, and two naturals $K, L \in \mathbb{N}$, decide if there exist $n_B \geq K$ distinct subsets $S_j \subseteq A$, such that, for all $j \in [n_B]$ we have $\sum_{a \in S_j} a \leq L$.

Proof (sketch). The MDP Γ that is constructed from an instance $A = \{a_1,\dots,a_n\}$, K, L of the k-th largest subset problem is drawn in Figure 1. We have $l_{\max} = n+1$. The construction is such that, if the sum of the weights visited from s_0 to s_n is at most L for a path π, then action β should be taken in s_n so that $f^{l_{max}}_{DirFixWMP}(\pi) = a_n + 1 - \frac{1}{n+1}$. Otherwise, action α should be taken, which leads to a direct fixed window mean-payoff equal to $a_n + 1$. In fact, we have that there exists at least K subsets of sum lower than or equal to L if and only if $\mathbb{E}^{\Gamma}_{s_0}(f^{l_{max}}_{DirFixWMP}) \leq \frac{1}{2^n}[(2^n - K) \cdot (a_n + 1) + K(a_n + 1 - \frac{1}{n+1})]$. □

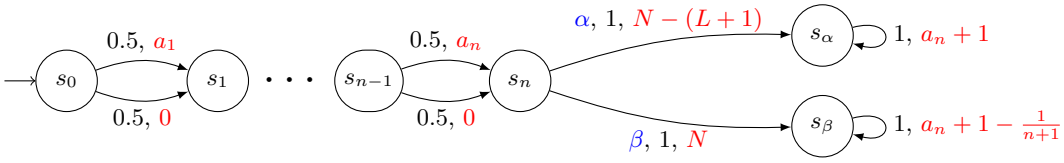

Fig. 1. The MDP that is constructed from the instance of the k-th largest subset problem $A = \{a_1, \ldots, a_n\}, K, L$. The probabilities appear in black, the weights in red. In the construction, we have $N = (n+1) \cdot (a_n + 1)$, where we assume that a_n is the maximum of all integer $(a_i)_{i \in [N]}$.

Since $l_{\max} = n + 1$, the reduction is still polynomial when $l_{\max}$ is given in unary. Thus, we cannot expect to have an algorithm that is polynomial in the value of l_{max} unless $P = PP$.

Expected value in MCs Consider a Markov chain $\mathcal{M} = \langle S, E, s_{init}, w, \mathbb{P} \rangle$. Let $f = f^{l_{max}}_{DirFixWMP}$. Then, for every path $\pi \in Paths^{\mathcal{M}}$, we have $f(\pi) \in \Lambda$ with $\Lambda = \{\frac{p}{q} \mid q \in [l_{max}],\ p \in [q \cdot W]_0\}$ and $W = \max_{e \in E} w(e)$. Then, if we denote the finite set Λ by the sequence of thresholds $\lambda_0 < \ldots < \lambda_n$, we have, for all $i \leq n-1$, $Pr(f^{-1}(\mathcal{M}, s_{init}, \lambda_i)) = Pr(f^{-1}(\mathcal{M}, s_{init}, [\lambda_i, \infty[)) - Pr(f^{-1}(\mathcal{M}, s_{init}, [\lambda_{i+1}, \infty[))$. Therefore, if $Pr(f^{-1}(\mathcal{M}, s_{init}, [\lambda_i, \infty[))$ is computed for all $1 \leq i \leq n$, the expected value $\mathbb{E}^{\mathcal{M}}_{s_{init}}(f)$ can be computed. For all $i \leq n$, we construct a new Markov chain $\mathcal{M}^{\lambda_i}_{l_{max}}$ so that the probability $Pr(f^{-1}(\mathcal{M}, s_{init}, [\lambda_i, \infty[))$ is equal to the probability of not reaching some state $(trap)$ in $\mathcal{M}^{\lambda_i}_{l_{max}}$. To do that, we first consider the inductive property of windows (see [7]).

Inductive property of windows. Let $\pi = s_0 \ldots \in Paths^{\mathcal{M}}$. Assume that there are $j \leq j' < n$ such that the window opened at s_j is still open at $s_{j'}$ and it is closed at $s_{j'+1}$ (with respect to λ_i). Then, any window opened between s_j to $s_{j'}$ (included) are also closed at $s_{j'+1}$ (with respect to λ_i).

This implies that we only have to remember the location of the largest window that is still opened, as well as the 'amount of payoff' that is required for it to be closed. If that window could not be closed within l_{max} steps, then the state $trap$ is reached. That is why, in the Markov chain $\mathcal{M}^{\lambda_i}_{l_{max}}$, the state space S' is equal to $S' = (S \times [l_{max} - 1]_0 \times [W \cdot (l_{max} - 1)]_0) \cup \{trap\}$. Then, we have the following lemma:

Lemma 6. *Let $1 \leq i \leq n$. Then:*

$$Pr((f)^{-1}(\mathcal{M}, s_{init}, [\lambda_i, \infty[)) = Pr(\pi \in \mathcal{M}^{\lambda_i}_{l_{max}} \mid \pi \models \neg\Diamond\{trap\})$$

Computing the probability of reaching some state in a Markov chain can be done in polynomial time. Moreover, we have that $|\mathcal{M}^{\lambda_i}_{l_{max}}| \leq |\mathcal{M}| \cdot l_{max} \cdot W \cdot l_{max} + 1$ and $|\Lambda| \leq l_{max} \cdot W \cdot l_{max}$. Hence, the theorem:

Theorem 9. *Computing the expected value of $f^{l_{\max}}_{DirFixWMP}$ in an MC $\mathcal{M}$ can be done in time $O(q_3(|S|, W, l_{max}))$ where q_3 is a polynomial function.*

If W and l_{max} are given in binary, the algorithm we have is pseudo-polynomial in W and l_{max}.

References

1. Baier, C., Katoen, J.: Principles of model checking. MIT Press (2008)
2. Bordais, B., Guha, S., Raskin, J.: Expected window mean-payoff. CoRR **abs/1812.09298** (2018)
3. Brázdil, T., Brozek, V., Chatterjee, K., Forejt, V., Kucera, A.: Two views on multiple mean-payoff objectives in markov decision processes. Logical Methods in Computer Science **10**(1) (2014). https://doi.org/10.2168/LMCS-10(1:13)2014, https://doi.org/10.2168/LMCS-10(1:13)2014
4. Brázdil, T., Chatterjee, K., Forejt, V., Kucera, A.: Trading performance for stability in markov decision processes. J. Comput. Syst. Sci. **84**, 144–170 (2017). https://doi.org/10.1016/j.jcss.2016.09.009, https://doi.org/10.1016/j.jcss.2016.09.009
5. Brázdil, T., Forejt, V., Kucera, A., Novotný, P.: Stability in graphs and games. In: 27th International Conference on Concurrency Theory, CONCUR 2016, August 23-26, 2016, Québec City, Canada. pp. 10:1–10:14 (2016)
6. Bruyère, V., Hautem, Q., Raskin, J.: On the complexity of heterogeneous multidimensional games. In: 27th International Conference on Concurrency Theory, CONCUR 2016, August 23-26, 2016, Québec City, Canada. LIPIcs, vol. 59, pp. 11:1–11:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016)
7. Chatterjee, K., Doyen, L., Randour, M., Raskin, J.: Looking at mean-payoff and total-payoff through windows. In: Automated Technology for Verification and Analysis - 11th International Symposium, ATVA 2013, Hanoi, Vietnam, October 15-18, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8172, pp. 118–132. Springer (2013)
8. Chatterjee, K., Doyen, L., Randour, M., Raskin, J.: Looking at mean-payoff and total-payoff through windows. Inf. Comput. **242**, 25–52 (2015). https://doi.org/10.1016/j.ic.2015.03.010, https://doi.org/10.1016/j.ic.2015.03.010
9. Ehrenfeucht, A., Mycielski, J.: Positional strategies for mean payoff games. International Journal of Game Theory **8**(2), 109–113 (Jun 1979). https://doi.org/10.1007/BF01768705, https://doi.org/10.1007/BF01768705
10. Filar, J., Vrieze, K.: Competitive Markov decision processes. Springer Science & Business Media (2012)
11. Haase, C., Kiefer, S.: The complexity of the kth largest subset problem and related problems. Inf. Process. Lett. **116**(2), 111–115 (2016). https://doi.org/10.1016/j.ipl.2015.09.015, https://doi.org/10.1016/j.ipl.2015.09.015
12. Hunter, P., Pérez, G.A., Raskin, J.: Looking at mean payoff through foggy windows. Acta Inf. **55**(8), 627–647 (2018). https://doi.org/10.1007/s00236-017-0304-7, https://doi.org/10.1007/s00236-017-0304-7
13. Karp, R.M.: A characterization of the minimum cycle mean in a digraph. Discrete Mathematics **23**, 309–311 (1978)
14. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., New York, NY, USA, 1st edn. (1994)
15. Simon, J.: On Some Central Problems in Computational Complexity. Ph.D. thesis, Ithaca, NY, USA (1975)
16. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM journal on computing **1**(2), 146–160 (1972)
17. Zwick, U., Paterson, M.: The complexity of mean payoff games on graphs. Theoretical Computer Science **158**(1-2), 343–359 (1996)