2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science

Multidimensional beyond worst-case and almost-sure problems for mean-payoff objectives

Lorenzo Clemente
Université Libre de Bruxelles
Brussels, Belgium

Jean-François Raskin
Université Libre de Bruxelles
Brussels, Belgium

***Abstract*—The *beyond worst-case threshold problem (BWC)*, recently introduced by Bruyère *et al.*, asks given a quantitative game graph for the synthesis of a strategy that i) enforces some minimal level of performance against any adversary, and ii) achieves a good expectation against a stochastic model of the adversary. They solved the BWC problem for finite-memory strategies and unidimensional mean-payoff objectives and they showed membership of the problem in NP∩coNP. They also noted that infinite-memory strategies are more powerful than finite-memory ones, but the respective threshold problem was left open.**

We extend these results in several directions. First, we consider multidimensional mean-payoff objectives. Second, we study both finite-memory and infinite-memory strategies. We show that the multidimensional BWC problem is coNP-complete in both cases. Third, in the special case when the worst-case objective is unidimensional (but the expectation objective is still multidimensional) we show that the complexity decreases to NP∩coNP. This solves the infinite-memory threshold problem left open by Bruyère *et al.*, and this complexity cannot be improved without improving the currently known complexity of classical mean-payoff games. Finally, we introduce a natural relaxation of the BWC problem, the *beyond almost-sure threshold problem (BAS)*, which asks for the synthesis of a strategy that ensures some minimal level of performance with probability one and a good expectation against the stochastic model of the adversary. We show that the multidimensional BAS threshold problem is solvable in P.

I. Introduction

In a two-player mean-payoff game played on a weighted graph [1], [2], given a threshold $v \in \mathbb{Q}$, we must decide if there exists a strategy for Player 1 (the controller) to force plays with mean-payoff values larger than v, against any strategy of Player 2 (the environment). In the *beyond worst-case threshold problem (BWC)*, recently introduced by Bruyère *et al.* in [3], we are additionally given a stochastic model for the *nominal*, i.e. expected, behaviour of Player 2. Then we are asked, given two threshold values $\mu, \nu \in \mathbb{Q}$, to decide if there exists a strategy for Player 1 that forces (i) plays with a mean-payoff value larger than μ against any strategy of Player 2, and (ii) an expected mean-payoff value larger than ν when Player 2 plays according to the stochastic model of his nominal behaviour. In the BWC problem, we thus need to solve simultaneously a two player zero-sum game for the worst-case and an optimization problem where the adversary has been replaced by a stochastic model of his behaviour.

BWC is a natural problem: in practice, we want to build systems that ensure good performances when the environment exhibits his nominal behaviour, and at the same time, that ensure some minimal performances no matter how the environment behaves. In [3], the BWC problem is solved for *finite-memory* strategies and *unidimensional* mean-payoff objectives, and shown to be in NP∩coNP. Also, it is noted there that *infinite-memory strategies* are more powerful than finite-memory ones, and that cannot even be approximated by the latter (already in the unidimensional case; cf. [4, Fig. 6] for an example). The respective threshold problem was left unsolved. We extend here these results in several directions.

A. Contributions

Our contributions are as follows. First, we consider d-*dimensional* mean-payoff objectives. Multiple dimensions are useful to model systems with *multiple objectives* that are potentially conflicting, and to analyze the possible trade-offs. For example, we may want to synthesize strategies that ensure a good QoS while keeping the energy consumption as low as possible. This extends the BWC problem with one additional level of conflicting trade-offs, which makes the analysis substantially harder. Second, we study both finite-memory and infinite-memory strategies. We show that the multidimensional BWC problem is coNP-complete in both cases, and so not more expensive than the plain multidimensional mean-payoff games. This is obtained as a coNP reduction to the solution of a linear system of inequalities of polynomial size. Correctness follows from non-trivial approximations results for finite/infinite-memory strategies inside end-components[1]. While in the unidimensional case optimal values for the expectation can always be achieved precisely (already by memoryless strategies), in our multidimensional setting this is not true anymore. To overcome this difficulty, we are able to show that achievable vectors can be approximated with arbitrary precision, which is sufficient for our analysis. Third, in the special case when the worst-case objective is unidimensional (but the expectation is still multidimensional), we show that the complexity decreases to NP∩coNP. This solves with optimal

Authors partially supported by the ERC inVEST (279499) project. The first author is partially supported by the Polish National Science Centre grant 2013/09/B/ST6/01575.

[1]Sub-MDPs which are strongly connected and closed w.r.t. the stochastic transitions.

1043-6871/15 $31.00 © 2015 IEEE
DOI 10.1109/LICS.2015.33

complexity the infinite-memory threshold problem left open in [3]. Finally, we introduce the *beyond almost-sure threshold problem (BAS)* which is a natural relaxation of the BWC problem. The BAS problem asks, given two threshold values $\vec{\mu}, \vec{\nu} \in \mathbb{Q}^d$, for the synthesis of a strategy for Player 1 that (i) ensures a mean-payoff larger than $\vec{\mu}$ almost surely, i.e. *with probability one*, and (ii) an expectation larger than $\vec{\nu}$ against the nominal behaviour of the environment. This problem has been independently considered (among other generalizations thereof) in [5]. We show that the multidimensional BAS threshold problem is solvable in P. As in the BWC problem, we reduce to a linear system of inequalities of polynomial size, but this time the reduction can be done in P.

B. Related works

Solutions to the expected unidimensional mean-payoff problem in Markov Decision Processes (MDP) can be found for example in [6], it can be solved in P, and pure memoryless strategies are sufficient to play optimally. The threshold problem for unidimensional mean-payoff games was first studied in [2], pure memoryless optimal strategies exist for both players, and the associated decision problem can be solved in NP∩coNP. As said above, BWC was introduced in [3] but studied only for finite memory strategies and unidimensional payoffs, the decision problem can be solve in NP∩coNP.

Multidimensional mean-payoff games are investigated in [7], [8], where it is shown that infinite-memory controllers are more powerful than finite-memory ones, and the finite-memory and general threshold problems are both coNP-complete. The expectation problem for multidimensional mean-payoff MDPs is in P, and finite-memory controllers always suffice [9]. Moreover, a recent study showed that one can add additional quantitative probability requirements for the mean-payoff to be above a certain threshold (while still optimizing the expectation), and that the resulting decision problem is P for the so-called *joint interpretation* (where the probability threshold is the same for all dimensions), and exponential for the *conjunction interpretation* (each dimension has a different probability threshold) [5] (cf. also [10]). In both cases, infinite-memory strategies are required to achieve the desired performance. Here, we study the *multidimensional mean-payoff BWC threshold problem*, for both finite-memory and arbitrary controllers. Our BWC threshold problem generalizes both the synthesis problem for multidimensional mean-payoff games and for multidimensional mean-payoff MDPs with no additional cost in worst-case computational complexity.

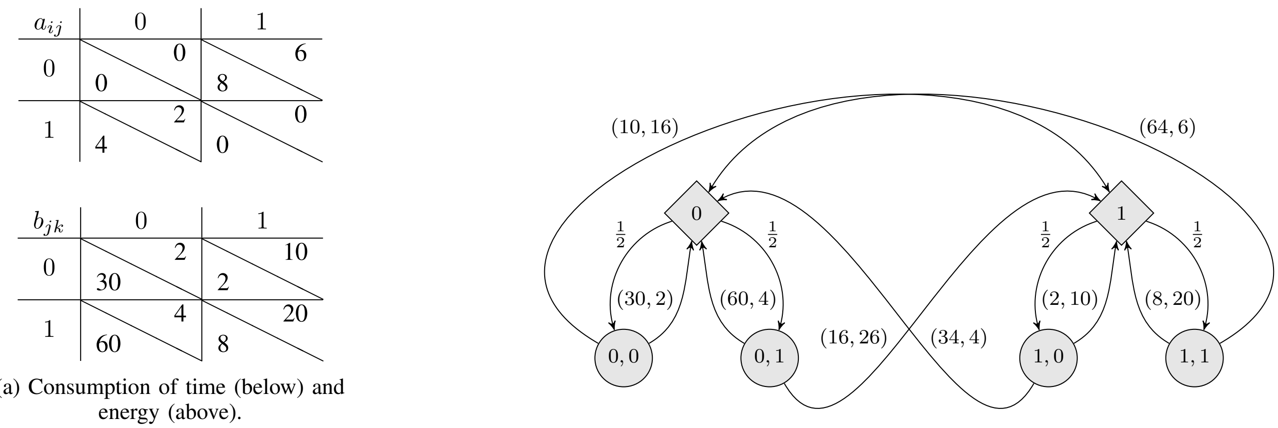
C. Illustrating example

Consider the following task system [11]: There are two configurations (0 and 1), and at each interaction between the controller and its environment, one new instance of two kind of tasks can be generated (0 and 1). The two tasks are generated with equal probability $1/2$ in the nominal behavior of the environment. Before serving pending task $k \in \{0,1\}$, the system may decide to go from configuration i to configuration j at cost a_{ij}, for $i, j \in \{0,1\}$, and then it has to serve the pending task k from the new configuration j at cost b_{jk}. Thus, the total cost is $a_{ij} + b_{jk}$. Costs are bidimensional: Each cost specifies an amount of time and energy; the actual parameters are shown in Fig. 1a. For example, from configuration 0, task 0 takes 30 time units to complete and it consumes 2 energy units, while from the other configuration the same task takes 2 time units and 10 energy units. We are interested in synthesizing controllers that optimize the expected/worst-case mean (i.e., per task) time and energy. There are trade-offs between the two measures: If the controller decides to serve a task quickly then the system consumes a large amount of energy, and vice versa. To analyze this example, we rephrase it as the multidimensional mean-payoff MDP depicted in Fig. 1b. For example, state 0 represents the fact that the system is in configuration 0 waiting for a task to arrive, while in state $(0,0)$ a task of the first type has arrived, and the controller needs to decide whether to serve it from the same configuration, or go to configuration 1. The objective of the controller is to guarantee worst-case mean time 24 under all circumstances, in that case the probabilities in the MDP are ignored and the probabilistic choice is replaced by an adversarial choice (we have thus a two-player zero sum game). Additionally, with the same strategy for the controller, we want to minimize the expected mean energy consumption in the nominal behaviour of the controller given by the stochastic model. If the controller decides to always serve tasks from configuration 0, then it ensures an expected mean energy consumption of 3, but under this strategy, the worst-case mean time is 60, which does not meet our worst-case objective of 24. A strategy for the controller that is good both for the worst-case and for the expectation can be obtained as follows: For two parameters $\alpha, \beta \in \mathbb{N}$, stay in configuration 0 for α consecutive tasks, then move to configuration 1 for β tasks, and then repeat. This ensures worst-case time $\frac{\alpha-1}{\alpha+\beta}60 + \frac{1}{\alpha+\beta}64 + \frac{\beta-1}{\alpha+\beta}8 + \frac{1}{\alpha+\beta}16$ and expected energy $\frac{\alpha-1}{\alpha+\beta}(\frac{1}{2}2 + \frac{1}{2}4) + \frac{1}{\alpha+\beta}(\frac{1}{2}4 + \frac{1}{2}6) + \frac{\beta-1}{\alpha+\beta}(\frac{1}{2}10 + \frac{1}{2}20) + \frac{1}{\alpha+\beta}(\frac{1}{2}16 + \frac{1}{2}26)$. By taking $\alpha = 1$ and $\beta = 3$, we obtain worst-case time 24 (thus meeting the requirement) and expected energy 14. Note the trade-off: To ensure a stronger guarantee on the mean time, we had to sacrifice the expected mean energy.

In this paper we address the problem of deciding the existence of controllers ensuring a worst-case (or almost-sure) threshold, while, at the same time, achieving a usually better expectation threshold under the nominal behavior of the environment. We consider the class of multidimensional mean-payoff objectives.

D. Structure of the paper

In Sec. II, we present the preliminaries that are necessary to define the BWC and BAS problems. In Sec. III, we solve the BWC problem both for finite and infinite memory strategies. In Sec. IV, we solve the BAS problem and show that finite memory strategies are sufficient to achieve the BAS threshold problem. Finally, in Sec.V we conclude with some final remarks. Full proofs can be found in the technical report [12].



(a) Consumption of time (below) and energy (above).

(b) Example of multidimensional mean-payoff MDP.

Fig. 1: Illustrating example.

II. Preliminaries

Let $\mathbb{N}$, $\mathbb{Q}$, and $\mathbb{R}$ be the set of natural, rational, and real numbers, respectively, and let $\mathbb{R}_{\pm\infty} = \mathbb{R} \cup \{+\infty, -\infty\}$. For two vectors $\vec{\mu}$ and $\vec{\nu}$ of the same dimension and a comparison operator $\sim \in \{\leq, <, >, \geq\}$, we write $\vec{\mu} \sim \vec{\nu}$ for the component-wise application of $\sim$. In particular, $\vec{\mu} > \vec{0}$ means that *every component* of $\vec{\mu}$ is strictly positive. A *probability distribution* on A is a function $R : A \to \mathbb{Q}^{\geq 0}$ s.t. $\sum_{a \in A} R(a) = 1$. The *support* of R is $\mathsf{Supp}(R) = \{a \in A \mid R(a) > 0\}$. Let $\mathcal{D}(A)$ be the set of probability distributions on A.

A. Weighted graphs

A *multi-weighted graph* is a tuple $G = (d, S, E, w)$, where $d \geq 1$ is the dimension, S is a finite set of states, $E \subseteq S \times S$ is the set of directed edges, and $w : E \to \mathbb{Z}^d$ is a function assigning to each edge a weight vector. When $d = 1$, we refer to G just as a weighted graph. With $\vec{x}[i]$ we denote the i-th component of a vector $\vec{x}$. For a state $s \in S$, let $E(s) = \{t \mid (s, t) \in E\}$ be its set of *successors*. We assume that each state s has at least one successor. Let W be the largest absolute value of a weight appearing in the graph.

A *play* in G is an infinite sequence of states $\pi = s_0 s_1 \cdots$ s.t. $(s_i, s_{i+1}) \in E$ for every $i \geq 0$. Let $\Omega^\omega_{s_0}(G)$ be the set of plays in G starting at s_0, and let $\Omega^\omega(G)$ be the set of all plays of G. When G is clear from the context, we omit it. The *prefix* of length n of a play $\pi = s_0 s_1 \cdots$ is the finite sequence $\pi(n) = s_0 s_1 \cdots s_{n-1}$. For a set of states $T \subseteq S$, Let $\Omega^*(G, T)$ be the set of prefixes of plays in G ending in a state $s_{n-1} \in T$.

The *total payoff* and *mean payoff* up to length n of a play $\pi = s_0 s_1 \cdots$ (or prefix of length at least n) are defined as $\mathsf{TP}_n(\pi) = \sum_{i=0}^{n-1} w(s_i, s_{i+1})$ and $\mathsf{MP}_n(\pi) = \frac{1}{n}\mathsf{TP}_n(\pi)$, respectively. The (lim-inf) total and mean payoffs on an infinite play π are then defined as $\mathsf{TP}(\pi) := \liminf_{n\to\infty} \mathsf{TP}_n(\pi)$ and $\mathsf{MP}(\pi) := \liminf_{n\to\infty} \mathsf{MP}_n(\pi)$.

B. Markov decision processes

A *Markov decision process*, or MDP, is a tuple $\mathcal{G} = (G, S^C, S^R, R)$, where $G = (d, S, E, w)$ is a multi-weighted graph, $\{S^C, S^R\}$ is a partition of S into states belonging to either the Controller player or to the Random player, respectively, and $R : S^R \to \mathcal{D}(S)$ is a function assigning a distribution over S to states belonging to Random s.t., for every $s \in S^R$, $\mathsf{Supp}(R(s)) = E(s)$. We do not allow $R(s)$ to assign probability zero to any successor of s.[2] Let Q be the largest denominator used to represent probabilities in R. We use Q as a measure of complexity for representing R.

In order to discuss the complexity of strategies for Controller, we represent them as *stochastic Moore machines*. A strategy for a MDP $\mathcal{G} = (G, S^C, S^R, R)$ is a tuple $f = (M, \alpha, f_u, f_o)$, consisting of a set of memory states M, the initial memory distribution $\alpha \in \mathcal{D}(M)$, the stochastic memory update function $f_u : S \times M \to \mathcal{D}(M)$, and the stochastic output function $f_o : S^C \times M \to \mathcal{D}(S)$, where $\mathsf{Supp}(f_o(s, m)) \subseteq E(s)$ for every $s \in S^C$ and $m \in M$. The update function is extended to sequences $f_u^* : S^* \to \mathcal{D}(M)$ inductively as $f_u^*(\varepsilon) = \alpha$ and $f_u^*(\pi s)(m') = \sum_{m \in M} f_u^*(\pi)(m) f_u(s, m)(m')$. The output function on sequences $f_o^* : S^* S^C \to \mathcal{D}(S)$ is defined as $f_o^*(\pi s)(s') = \sum_{m \in M} f_u^*(\pi)(m) f_o(s, m)(s')$. A play $\pi = s_0 s_1 \cdots$ is *consistent* with a Controller's strategy f if, and only if, for every i s.t. $s_i \in S^C$, we have $s_{i+1} \in \mathsf{Supp}(f_o^*(s_0 s_1 \cdots s_i))$. Given a state s_0 and a Controller's strategy f, the set of *outcomes* $\Omega^f_{s_0}(G)$ is the set of plays starting at s_0 which are consistent with f.

A strategy f is *pure* iff $\mathsf{Supp}(f_u(s, m))$ and $\mathsf{Supp}(f_o(s, m))$ are both singletons. A strategy f is *memoryless* iff $|M| = 1$, *finite-memory* iff $|M| < \infty$, and *infinite-memory* iff $|M| = \infty$.

[2]This restriction will simplify the presentation. It is not present in [3], but it can be easily lifted.

Let $\Delta(\mathcal{G})$, $\Delta_{\mathrm{P}}(\mathcal{G})$, $\Delta_{\mathrm{F}}(\mathcal{G})$, and $\Delta_{\mathrm{PF}}(\mathcal{G})$ be the sets of all, resp., pure, finite-memory, and pure finite-memory strategies.

C. Markov chains

A *Markov chain* is an MDP where no state belongs to Controller, i.e., $S^C = \emptyset$, and in this case we just write $\mathcal{G} = (G, R)$. An *event* is a measurable set of plays $A \subseteq \Omega^\omega(G)$. Given a state s_0 and an event $A \subseteq \Omega^\omega_{s_0}(G)$, let $\mathbb{P}^{\mathcal{G}}_{s_0}[A]$ be the probability that a play starting in s_0 belongs to A, which exists and it is unique by Carathéodory's extension theorem [13]. An even is *almost sure* if it has probability 1. For a measurable payoff function $v : \Omega^\omega(G) \to \mathbb{R}^d_{\pm\infty}$, let $\mathbb{E}^{\mathcal{G}}_{s_0}[v]$ be the expected value of v of a play starting in s_0.

A Markov chain $\mathcal{G}$ is *unichain* if it contains exactly one bottom strongly connected component (BSCC). Therefore, if $\mathcal{G}$ is unichain, then all states in its unique BSCC are visited infinitely often almost surely, and the mean payoff equals its expected value almost surely.

Given a MDP $\mathcal{G} = (G, S^C, S^R, R)$ and a strategy f for Controller represented as the stochastic Moore machine (M, α, f_u, f_o), let the *induced Markov chain* be $\mathcal{G}[f] = (G', R')$, where $G' = (d, S \times M, w', E')$ with $((s, m), (s', m')) \in E'$ iff $(s, s') \in E$, $m' \in \mathsf{Supp}(f_u(s, m))$, and $s' \in \mathsf{Supp}(f_o(s, m))$ whenever $s \in S^C$, $w'((s, m), (s', m')) = w(s, s')$ for every $((s, m), (s', m')) \in E'$, $R'(s, m)(s', m') = R(s)(s') \cdot f_u(s, m)(m')$ for every $s \in S^R$, and $R'(s, m)(s', m') = f_o(s, m)(s') \cdot f_u(s, m)(m')$ for every $s \in S^C$. Note that $\mathcal{G}[f]$ is finite iff f is finite-memory. By a slight abuse of terminology, we say that a strategy f is *unichain* if $\mathcal{G}[f]$ is unichain. Plays in $\mathcal{G}[f]$ can be mapped to plays in $\mathcal{G}$ by a projection operator $\mathsf{proj}(\cdot) : \Omega^\omega(G') \to \Omega^\omega(G)$ which discards the memory of f. Given a state s_0, a Controller's strategy f, and an event $A \subseteq \Omega^\omega_{s_0}$, let $\mathbb{P}^{\mathcal{G}}_{s_0, f}[A] := \mathbb{P}^{\mathcal{G}[f]}_{s_0}\left[\mathsf{proj}^{-1}(A)\right]$. For a measurable payoff function $v : \Omega^\omega(G) \to \mathbb{R}^d_{\pm\infty}$, let $\mathbb{E}^{\mathcal{G}}_{s_0, f}[v] := \mathbb{E}^{\mathcal{G}[f]}_{s_0}[v']$, where $v'(\pi) := v(\mathsf{proj}(\pi))$.

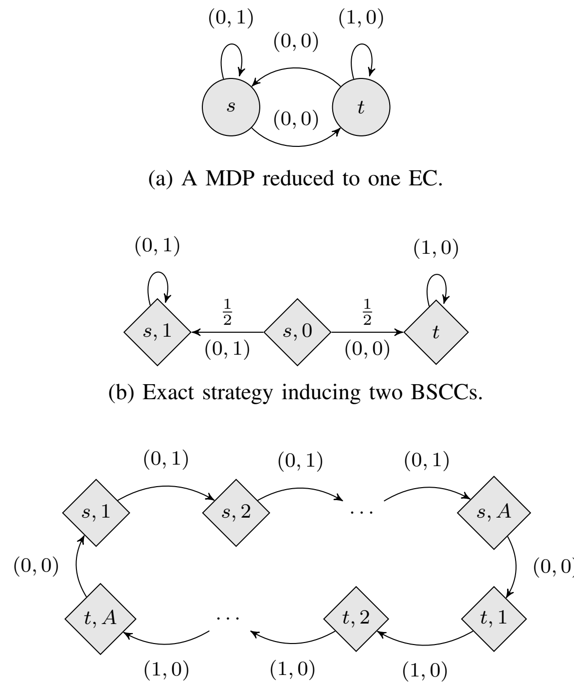
D. End-components

An *end-component* (EC) of a MDP $\mathcal{G}$ is a set of states $U \subseteq S$ s.t. a) the induced sub-graph $(U, E \cap U \times U)$ is strongly-connected, and b) for any stochastic state $s \in U \cap S^R$, $E(s) \subseteq U$. Thus, Controller can surely keep the game inside an EC, and almost surely visits all states therein. For an end-component U of $\mathcal{G}$, we denote by $\mathcal{G} \downharpoonright U$ the MDP obtained by restricting $\mathcal{G}$ to U in the natural way. ECs are central in the analysis of MDPs thanks to the following result.

Proposition 1 (cf. [14])**.** *For any Controller's strategy $f \in \Delta(\mathcal{G})$, the set of states visited infinitely often when playing according to f is almost surely an EC.*

E. Expected-value objective

For a MDP $\mathcal{G}$, a starting state s_0, and Controller's strategy $f \in \Delta(\mathcal{G})$, the set of *expected-value achievable solutions for f* is $\mathsf{ExpSol}^+_{\mathcal{G}}(s_0, f) = \{\vec{\nu} \in \mathbb{R}^d \mid \mathbb{E}^{\mathcal{G}}_{s_0, f}[\mathsf{MP}] > \vec{\nu}\}$, i.e., it is the set of vectors $\vec{\nu}$ s.t. Controller can guarantee an expected mean payoff $> \vec{\nu}$ from state s_0 by playing f. The set of *expected-value achievable solutions* is $\mathsf{ExpSol}^+_{\mathcal{G}}(s_0) = \bigcup_{f \in \Delta(\mathcal{G})} \mathsf{ExpSol}^+_{\mathcal{G}}(s_0, f)$. Given a state s_0 and rational threshold vector $\vec{\nu} \in \mathbb{Q}^d$, the *expected-value threshold problem* asks whether $\vec{\nu} \in \mathsf{ExpSol}^+_{\mathcal{G}}(s_0)$.



(a) A MDP reduced to one EC.

(b) Exact strategy inducing two BSCCs.

(c) Approximate finite-memory strategy inducing one BSCC.

Fig. 2: Approximating the expectation inside ECs.

Theorem 1 ([9])**.** *The expected-value threshold problem for multidimensional mean-payoff MDPs is in* P.

While randomized finite-memory strategies are both necessary and sufficient in general for achieving a given expected mean payoff, in ECs we can use randomized finite-memory *unichain* strategies to *approximate* achievable vectors. Being unichain ensures that the mean payoff equals the expectation almost surely. By standard convergence results in Markov chains, this entails that by playing such a strategy for sufficiently long time we obtain an average mean payoff close to the expectation with high probability. We crucially use this property in the constructions leading to the main results of Sec. III and IV; cf. Lemmas 3, 5, and 8.

Example 1. We illustrate the idea in the single end-component MDP in Fig. 2a (cf. [8, Fig. 3]). There exists a simple randomized 2-memory strategy f achieving expected mean payoff precisely $(\frac{1}{2}, \frac{1}{2})$ which decides, with equal probability, whether to stay forever in s or in t. However, the induced Markov chain has two BSCCs; cf. Fig. 2b. While intuitively no pure finite-memory strategy can achieve mean payoff exactly equal $(\frac{1}{2}, \frac{1}{2})$ in this example, finite-memory unichain strategies can approximate this value. For a parameter $A \in \mathbb{N}$, consider the strategy g_A which stays in s for A steps, and then goes to t, stays in t for A steps, and then goes back to s, and repeats this scheme forever. The induced Markov chain has only one BSCC, thus g_A is unichain; cf. Fig. 2c. The strategy g_A achieves expected (and worst-case) mean payoff $\left(\frac{A}{2A+2}, \frac{A}{2A+2}\right)$, which converges from below to $(\frac{1}{2}, \frac{1}{2})$ as $A \to \infty$.

Lemma 1. *Let $\mathcal{G}$ be a multidimensional mean-payoff MDP, let s_0 be a state in an EC U thereof, and let $\vec{\nu} \in \mathsf{ExpSol}^+_{\mathcal{G}\downharpoonright U}(s_0)$ be an expectation vector achievable by remaining inside U. There exists a* finite-memory unichain *strategy $g \in \Delta_F(\mathcal{G})$ achieving the same expectation $\vec{\nu} \in \mathsf{ExpSol}^+_{\mathcal{G}\downharpoonright U}(s_0, g)$.*

Remark 1. In the lemma above, we can take g to be even a *pure* finite-memory unichain strategy. This can be obtained by a de-randomization technique at the cost of introducing extra memory of size exponential in the number of the states controlled by the player. However, we do not need this stronger result in the rest of the paper, and we content ourselves with randomized strategies for simplicity.

Proof sketch. By the results of [9], there exists a randomized finite-memory strategy f achieving expected mean payoff $\vec{\nu}^* > \vec{\nu}$ which surely stays inside U. However, $(\mathcal{G} \downharpoonright U)[f]$ is not unichain in general. By Proposition 1, the set of states visited infinitely often by a play in $(\mathcal{G} \downharpoonright U)[f]$ is an EC almost surely. Since there are finitely many different ECs, there are probabilities $\alpha_1, \ldots, \alpha_n > 0$ and ECs $U_1, \ldots, U_n \subseteq U$ s.t. the set of states visited infinitely often by a play in $(\mathcal{G} \downharpoonright U)[f]$ is U_1 with probability α_1, …, U_n with probability α_n. By Proposition 1, $\alpha_1 + \cdots + \alpha_n = 1$. In the first step, we define a "local" randomized memoryless strategy g_i which plays as f once inside U_i. No approximation is introduced in this step. In the second step, we combine the local randomized memoryless strategies g_i's above. We build a randomized finite-memory strategy g which cycles between $U_1, \ldots, U_n$ and plays according to g_i inside each U_i a fraction $\approx \alpha_i$ of the time. This is possible since U_i, U_j are almost surely mutually inter-reachable due to the fact that we are always inside the EC U. By construction, $(\mathcal{G} \downharpoonright U)[g]$ is unichain since g cycles between all the ECs $U_1, \ldots, U_n$. Moreover, for every $\varepsilon > 0$, we can make the expected fraction of time spent changing component smaller than ε. Thus, g achieves expected mean payoff at least $(1-\varepsilon) \cdot \vec{\nu}^* - (W, \ldots, W) \cdot \varepsilon$, where W is the largest absolute value of any weight in $\mathcal{G}$. The latter quantity can be made $> \vec{\nu}$ for sufficiently small $\varepsilon > 0$. □

F. Worst-case objective

For a MDP $\mathcal{G}$, a starting state s_0, and a Controller's strategy $f \in \Delta(\mathcal{G})$, the set of *worst-case achievable solutions for f* is defined as $\mathsf{WCSol}^+_{\mathcal{G}}(s_0, f) = \{\vec{\mu} \in \mathbb{R}^d \mid \forall \pi \in \Omega^f_{s_0} \cdot \mathsf{MP}(\pi) > \vec{\mu}\}$, i.e., it is the set of vectors $\vec{\mu}$ s.t. Controller can surely guarantee a mean payoff $> \vec{\mu}$ from state s_0 by playing f. The set of *worst-case achievable solutions* is $\mathsf{WCSol}^+_{\mathcal{G}}(s_0) = \bigcup_{f \in \Delta(\mathcal{G})} \mathsf{WCSol}^+_{\mathcal{G}}(s_0, f)$. Given a state s_0 and rational threshold vector $\vec{\mu} \in \mathbb{Q}^d$, the *worst-case threshold problem* asks whether $\vec{\mu} \in \mathsf{WCSol}^+_{\mathcal{G}}(s_0)$.

With this worst-case interpretation, the randomized choices in the MDP are replaced by purely adversarial ones, and the MDP can thus be viewed as a two-player zero-sum game. While infinite-memory strategies are more powerful than finite-memory ones for the worst-case objective, the latter suffice to approximate achievable vectors. We make extensive use of this property in Sec. III-A where we restrict our attention to finite-memory strategies.

Lemma 2 (cf. Lemma 15 of [8])**.** *Let $\mathcal{G}$ be a multidimensional mean-payoff MDP, s_0 a state therein, and let $\vec{\mu} \in \mathsf{WCSol}^+_{\mathcal{G}}(s_0)$. There exists a pure* finite-memory *strategy $f \in \Delta_{PF}(\mathcal{G})$ for Controller s.t. $\vec{\mu} \in \mathsf{WCSol}^+_{\mathcal{G}}(s_0, f)$.*

The finite-memory strategy threshold problem for multidimensional mean-payoff games is coNP-complete [7], [8]. By the lemma above, finite memory controllers suffice in our setting, and we obtain the following complexity characterization.

Theorem 2 ([7], [8])**.** *The worst-case threshold problem for multidimensional mean-payoff MDPs is* coNP*-complete.*

In the unidimensional case, memoryless strategies suffice for both players [1], [2], and the complexity is NP∩coNP (and even UP∩coUP [11], [15]). It is open since long time whether this problem is in P.

Theorem 3 ([1], [2])**.** *The worst-case threshold problem for unidimensional mean-payoff MDPs is in* NP∩coNP.

III. Beyond worst-case synthesis

We generalize [3] to the multidimensional setting. Given a MDP $\mathcal{G}$, a starting state s_0, and a Controller's strategy f, the set of *beyond worst-case achievable solutions for f*, denoted $\mathsf{BWCSol}^+_{\mathcal{G}}(s_0, f)$, is the set of pairs of vectors $(\vec{\mu}; \vec{\nu}) \in \mathbb{R}^{2d}$ s.t. f surely guarantees a worst-case mean payoff $> \vec{\mu}$, and achieves an expected mean payoff $> \vec{\nu}$ starting from s_0,

$$\mathsf{BWCSol}^+_{\mathcal{G}}(s_0, f) = \left\{ (\vec{\mu}; \vec{\nu}) \in \mathbb{R}^{2d} \;\middle|\; \begin{array}{c} \vec{\mu} \in \mathsf{WCSol}^+_{\mathcal{G}}(s_0, f) \\ \text{and} \\ \vec{\nu} \in \mathsf{ExpSol}^+_{\mathcal{G}}(s_0, f) \end{array} \right\}$$

Let $\mathsf{BWCSol}^+_{\mathcal{G}}(s_0) = \bigcup_{f \in \Delta(\mathcal{G})} \mathsf{BWCSol}^+_{\mathcal{G}}(s_0, f)$ be the set of *beyond worst-case achievable solutions*. Given a starting state s_0 and a pair of threshold vectors $(\vec{\mu}; \vec{\nu}) \in \mathbb{R}^{2d}$, the *beyond worst-case threshold problem* (BWC) asks whether $(\vec{\mu}; \vec{\nu}) \in \mathsf{BWCSol}^+_{\mathcal{G}}(s_0)$.

Remark 2. We assume w.l.o.g. that $\vec{\mu} = \vec{0}$. This follows by shifting each component by an appropriate amount. We further assume w.l.o.g. that $\vec{\nu} \geq \vec{0}$. This follows from the fact that, since the mean payoff is surely $> \vec{0}$ by the worst-case objective, then also the expectation is $> \vec{0}$.

Remark 3. We say that $\mathcal{G}$ is *pruned* if $\vec{0} \in \mathsf{BWCSol}^+_{\mathcal{G}}(s)$ for every state s therein. Controller cannot satisfy the BWC objective if she ever visits a state s not satisfying the worst-case objective. Many of our results are thus stated under the condition that $\mathcal{G}$ is pruned. However, pruning an MDP, i.e., removing those states which are losing w.r.t. the worst-case

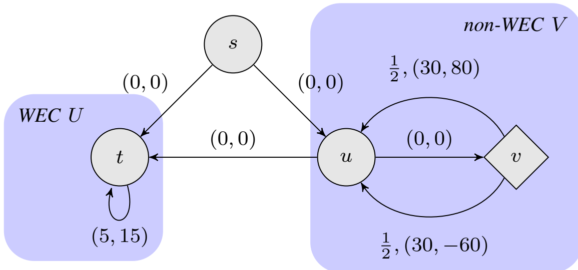


Fig. 3: Running example.

objective, requires solving a mean-payoff game, and this will have a crucial impact on the complexity.

The finite-memory threshold problem for the unidimensional beyond worst-case problem has been studied in [3].

Theorem 4 ([3])**.** *The finite-memory threshold problem for the unidimensional beyond worst-case problem for mean-payoff objectives is in* NP∩coNP.

A. Finite-memory synthesis

In this section, we address the problem of deciding whether there exists a finite-memory strategy for the BWC problem in the multidimensional setting. By Proposition 1, we know that the set of states visited infinitely often by any strategy (not necessarily a finite-memory one) is almost surely an EC. The crucial observation is that, when restricted to finite-memory, the same holds for ECs of a special kind. An EC U is *winning* (WEC) iff Controller can surely guarantee the worst-case threshold $> \vec{0}$ when constrained to remain in U, starting from any state therein. Whether a EC is winning depends on the worst-case objective alone.

The following proposition is central in the analysis of the BWC problem for finite-memory strategies; cf. [3, Lemma 4] in the unidimensional case.

Proposition 2. *Let f be a* finite-memory *strategy satisfying the worst-case threshold problem. The set of states visited infinitely often under f is almost surely a* winning *EC.*

Running example. As a simple example that will be used through the rest of the paper, consider the MDP in Fig. 3. There are only two ECs U and V, of which U is winning, but V is not. Indeed, from v the adversary can always select the lower edge with payoff $(30, -60)$. In U we can achieve expectation $(5, 15)$, and from V we can achieve expectation $(15, 5)$. Therefore, according to the lemma above, any finite-memory strategy satisfying the worst-case objective will eventually go to U almost surely.

We proceed by analyzing WECs separately in Sec. III-A1, and then we tackle general MPDs in Sec. III-A2. This will yield our complexity result in Sec. III-A3.

1) Inside a WEC: We show that inside WECs finite-memory strategies always suffice for the BWC objective. In particular, the threshold problem in WECs immediately reduces to an expectation threshold problem.

Lemma 3. *Let $\mathcal{G}$ be a pruned multidimensional mean-payoff MDP, let s_0 be a state in a WEC W of $\mathcal{G}$, and let $\vec{\nu} \in \mathsf{ExpSol}^{+}_{\mathcal{G}\downharpoonright W}(s_0)$ with $\vec{\nu} \geq \vec{0}$ be an expectation achievable by remaining inside W. There exists a* randomized finite-memory *strategy $h \in \Delta_F(\mathcal{G})$ s.t. $(\vec{0}; \vec{\nu}) \in \mathsf{BWCSol}^{+}_{\mathcal{G}\downharpoonright W}(s_0, h)$ that also remains inside W.*

Remark 4. The statement of the lemma holds even with h a *pure* finite-memory strategy, by applying Remark 1 when constructing the expectation strategy which is part of h. However, randomized strategies suffice for our purposes.

We use finite-memory strategies defined in WECs (such as h above) when constructing a global BWC strategy in the analysis of arbitrary MDPs in Sec. III-A2. The construction of h is done in a way analogous to the proof of Theorem 5 in [3]; cf. the technical report for the details [12]. However, the analysis in the multidimensional case is considerably more difficult than in previous work. It crucially relies on Lemma 1 for the extraction of *finite-memory unichain* strategies approximating the expectation objective inside ECs. Note that in the unidimensional case of [3] optimal expectation values can be reached exactly already by *pure memoryless unichain strategies* (no approximation needed). This is an key technical difference between our multidimensional setting and the unidimensional one of [3].

2) General case: We reduce the finite-memory BWC problem to the solution of a system of linear inequalities. This is similar to the solution of the multidimensional expectation problem presented in [9]. When only the expectation is considered, the intuition is that a "global expectation" is obtained by combining together "local expectations" achieved in ECs. Thus, a strategy for the expectation works in two phases:

Phase I: Reach ECs with appropriate probabilities.

Phase II: Once inside an EC, switch to a local expectation strategy to achieve the right "local expectation".

In the BWC problem, we need to enforce two extra conditions: First, only "local expectations" from *winning* ECs should be considered (by Proposition 2 finite-memory controllers cannot stay in a non-WEC forever with non-zero probability). Second, "local expectations" should be $> \vec{0}$ in order to satisfy the worst-case objective (a negative "local expectation" would violate the worst-case objective). Accordingly, a strategy for the BWC problem behaves as follows:

Phase I: Reach WECs with appropriate probabilities.

Phase II: Once inside a WEC, switch to a local BWC strategy to achieve the right "local expectation" $> \vec{0}$.

We write a system of linear inequalities expressing this two-phase decomposition. W.l.o.g. we assume that state s_0 belongs to Controller, and that all WECs are reachable with positive probability from s_0 (unreachable states can be removed). Consider the system T in Fig. 4. For each state $s \in S$ we have a variable y_s, and for each edge $(s, t) \in E$ we have

$$1_{s_0}(s) + \sum_{(r,s)\in E} y_{rs} = \sum_{(s,t)\in E} y_{st} + y_s \qquad \forall s \in S \tag{A1}$$

$$y_{st} = R(s)(t) \cdot (\sum_{(r,s)\in E} y_{rs} - y_s) \qquad \forall (s,t) \in E \text{ with } s \in S^R \tag{A1'}$$

$$\sum_{\text{MWEC } U} \sum_{s\in U} y_s = 1 \tag{A2}$$

$$\sum_{s\in U} y_s = \sum_{(r,s)\in E\cap U\times U} x_{rs} \qquad \forall \text{ MWEC } U \tag{B}$$

$$\sum_{(r,s)\in E} x_{rs} = \sum_{(s,t)\in E} x_{st} \qquad \forall s \in S \tag{C1}$$

$$x_{st} = R(s)(t) \cdot \sum_{(r,s)\in E} x_{rs} \qquad \forall (s,t) \in E \text{ with } s \in S^R \tag{C1'}$$

$$\sum_{(s,t)\in E} x_{st} \cdot w(s,t)[i] > \vec{\nu}[i] \qquad \forall (1 \le i \le n) \tag{C2}$$

$$\sum_{(s,t)\in E\cap U\times U} x_{st} \cdot w(s,t)[i] > 0 \qquad \forall \text{ MWEC } U, 1 \le i \le n \tag{C3}$$

Fig. 4: Linear system T for the BWC finite-memory threshold problem.

variables x_{st} and y_{st}. System T can be divided into three parts. The first part consists of Equations (A1)–(A2). Variable y_s represents the probability that, upon visiting state s, we switch to Phase II. Variables y_{st}'s are used to express flow conditions. In Eq. (A1) we put an initial flow of 1 in s_0, and we require that the total incoming flow to a state equals the outgoing flow (including the leak y_s). In Eq. (A1') ensures that the outgoing flow through an edge y_{st} from a stochastic state s is a fixed fraction of the incoming flow. Finally, Eq. (A2) states that we switch to Phase II in a WEC almost surely.

Before explaining the other two parts of T, we need to introduce maximal WECs. A *maximal WECs* (MWEC) is a WEC which is not strictly included into another WEC. The restriction to MWECs is crucial for complexity. The second part of T consists of Eq. (B) and it provides a link between Phase I and Phase II. Variable x_{st} represents the long-run frequency of edge (s,t). Eq. (B) links the transient behaviour before switching inside a certain MWEC and the steady state behaviour once inside it. More precisely, it guarantees that the probability to switch inside a certain MWEC equals the total long-run frequency of all edges in the MWEC.

Finally, the remaining equations make up the third part of T. Eq. (C1) is a flow condition for the x_{st}'s, stating that the incoming flow to a state equals the outgoing flow. Eq. (C1') forces the flow to respect the probabilities of stochastic states. Eq. (C2) guarantees that the expected mean payoff is $> \vec{\nu}$, as required. Eq. (C3) needs some justification. It is specific to our setting and it does not follow from [9]. This equation specifies that the expected mean payoff is $> \vec{0}$ *inside every MWECs*. We need to ensure that only "local" expected mean payoffs $> \vec{0}$ should be considered in WECs, in order to be able to apply the results from the previous Sec. III-A1. Eq. (C3) imposes a seemingly strong constraint by requiring that *all WECs* are visited infinitely often with positive probability. Ideally, we would like to guess which are the MWECs which need to be visited infinitely often with positive probability, but this would not yield a good complexity, since there are exponentially many different sets of MWECs. Instead, we require that *every* MWEC is visited infinitely often with some positive probability. Since we are only interested in approximating the expectation, it is always possible to put an arbitrary small total probability on MWECs that do not contribute to the "global" mean payoff. This is formalized below.

Proposition 3. *Let $\mathcal{G}$ be a pruned multidimensional mean-payoff MDP. If there exists a finite-memory strategy h s.t. $(\vec{0};\vec{\nu}) \in \mathsf{BWCSol}^{+}_{\mathcal{G}}(s_0, h)$, then there exists a finite-memory strategy h^* with the same property, and such that, for every MWEC U, the set of states visited infinitely often by h^* is a subset of U with positive probability.*

Proof. Since by assumption all MWEC are reachable with positive probability from s_0, for every MWEC U there exists a strategy f_U reaching U with positive probability from s_0. Moreover, since U is a WEC, there exists a strategy f^{wc}_U for the worst-case objective $> \vec{0}$ that surely remains in U. Let f^{wc} be a worst-case strategy winning everywhere (it exists since $\mathcal{G}$ is pruned by assumption). We construct the following strategy f_N parametrized by a natural number $N > 0$:

- Choose a MWEC U uniformly at random.
- Play f_U for N steps.
  - If after N steps the play is in U, then switch to f^{wc}_U.
  - Otherwise, switch to f^{wc}.

By construction f_N is winning for the worst-case for every $N > 0$. Moreover, it is easy to see that there exists an N^* sufficiently large s.t., for every MWEC U, f_{N^*} visits U infinitely often with positive probability.

Finally, the strategy h^* plays with probability $p > 0$ according to f_{N^*}, and otherwise according to h. Since both f_{N^*} and h are winning for the worst-case, so it is h^*. The expected mean payoff of h^* converges from below to the expected mean payoff of h for $p > 0$ sufficiently small. Therefore, there exists $p > 0$ s.t. $(\vec{0}; \vec{\nu}) \in \mathsf{BWCSol}^+_{\mathcal{G}}(s_0, h^*)$. ☐

We now state the correctness of the reduction.

Lemma 4. *Let $\mathcal{G}$ be a pruned multidimensional mean-payoff MDP, let $s_0 \in S$, and let $\vec{\nu} \geq \vec{0}$. There exists a* finite-memory *strategy h s.t. $(\vec{0}; \vec{\nu}) \in \mathsf{BWCSol}^+_{\mathcal{G}}(s_0, h)$, if, and only if, the system T has a non-negative solution.*

The rest of this section is devoted to the proof of the lemma above. Both directions are non-trivial. For the right-to-left direction, we need to explain which kind of strategies can be extracted from a non-negative solution of T. The following lemma shows that from a non-negative solution of T we can extract a strategy for the expectation combining only "local mean payoffs" $> \vec{0}$ and visiting infinitely often each MWEC with positive probability.

Proposition 4. *If T has a non-negative solution, then there exists a finite-memory strategy $\hat{h}$ s.t. $\vec{\nu} \in \mathsf{ExpSol}^+_{\mathcal{G}}(s_0, \hat{h})$, and*

1) *For every MWEC U, there is a probability $y^*_U > 0$ s.t. the set of states visited infinitely often by $\hat{h}$ is a subset of U with probability y^*_U.*
2) *Once $\hat{h}$ reaches the MWEC U, it achieves expected mean payoff $\vec{\nu}_U > \vec{0}$.*
3) *$\sum_{MWEC\ U} y^*_U \cdot \vec{\nu}_U > \vec{\nu}$.*

Proof. Let $\{y^*_s\}_{s \in S}$, $\{y^*_{st}\}_{(s,t) \in E}$, and $\{x^*_{st}\}_{(s,t) \in E}$ be a non-negative solution to T. Proposition 4.2 of [9] essentially shows how to construct from the solution above a finite-memory strategy $\hat{h}$ s.t. $\vec{\nu} \in \mathsf{ExpSol}^+_{\mathcal{G}}(s_0, \hat{h})$.

For a MWEC U, let

$$y^*_U = \sum_{s \in U} y^*_s \tag{1}$$

By Eq. (C3), for every MWEC U there exist $s, t \in U$ s.t. $x^*_{st} > 0$. Together with Eq. (B), this implies that $y^*_U > 0$, which proves Point 1.

For a MWEC U, let

$$\vec{\nu}_U = \sum_{(s,t) \in E \cap U \times U} x^*_{st} \cdot w(s, t) \tag{2}$$

and notice that $\vec{\nu}_U$ is the expected mean payoff of $\hat{h}$ once inside U. By Eq. (C3), $\vec{\nu}_U > \vec{0}$, which proves Point 2.

Eq. (B) implies that $\hat{h}$ eventually stays forever inside a WEC almost surely. Consequently, $\sum_{\text{MWEC}\ U} y^*_U = 1$. Since states visited infinitely often with probability zero do not contribute to the expected mean payoff, it suffices to look at MWECs. By the prefix independence of the mean payoff value function, and since MWEC U is reached with probability y^*_U, strategy $\hat{h}$ achieves expected mean payoff $\sum_{\text{MWEC}U} y^*_U \cdot \vec{\nu}_U$. By Point 1), the latter quantity is $> \vec{\nu}$. ☐

We are now ready to prove Lemma 4.

Proof of Lemma 4. For the left-to-right direction, assume that h is a finite-memory strategy guaranteeing $(\vec{0}; \vec{\nu}) \in \mathsf{BWCSol}^+_{\mathcal{G}}(s, h)$. Proposition 4.4 of [9] essentially shows that any strategy satisfying the expectation objective $> \vec{\nu}$ induces a solution to T satisfying Equations (A1)–(C2), except that Eq. (B) should be interpreted over MECs (instead of MWECs). (This follows from the fact that the set of states visited infinitely often by any strategy is an EC almost surely; cf. Proposition 1.) However, since $\vec{0} \in \mathsf{WCSol}^+_{\mathcal{G}}(s, h)$ and h is finite-memory, we can apply Proposition 2 and deduce that h visits infinitely often a *winning* EC almost surely. Thus Eq. (B) is satisfied even over MWECs.

It remains to address Eq. (C3). By Proposition 3, there exists a strategy h^* s.t., for every MWEC U, h^* eventually stays forever in U with a positive probability. This implies that, when constructing a solution to T induced by h^* (as above), for every MWEC U and $s, t \in U$, $x^*_{st} > 0$. Moreover, since h^* is winning for the worst-case, it achieves an expected mean payoff $> \vec{0}$ in U, and thus Eq. (C3) is satisfied.

For the right-to-left direction, assume that T has a non-negative solution. Let $\hat{h}$ be the strategy in $\mathcal{G}$ given by Proposition 4. For every MWEC U, let y^*_U and $\vec{\nu}_U$ be as given in the statement of the proposition. While $\hat{h}$ alone is not sufficient to show $(\vec{0}; \vec{\nu}) \in \mathsf{BWCSol}^+_{\mathcal{G}}(s)$ since it does not satisfy the worst-case objective in general, we show how to construct from it another finite-memory strategy h^{cmb} ensuring the BWC objective. The latter strategy is obtained by combining together the following strategies:

- Let h^{wc} be a finite-memory strategy in $\mathcal{G}$ ensuring the worst-case mean payoff $\vec{0} \in \mathsf{WCSol}^+_{\mathcal{G}}(t, h^{wc})$ from every state t in $\mathcal{G}$. This is possible since $\mathcal{G}$ is pruned.
- For each MWEC U, let h_U be a finite-memory strategy s.t. $(\vec{0}; \vec{\nu}_U) \in \mathsf{BWCSol}^+_{\mathcal{G}}(t, h_U)$ for every state $t \in U$. This strategy can be obtained as follows. Let $\mathcal{G} \downharpoonleft U$ be the game $\mathcal{G}$ restricted to the EC U. By Point 2 of Proposition 4, $\vec{\nu}_U \in \mathsf{ExpSol}^+_{\mathcal{G} \downharpoonleft U}(t_0, h_U)$ for some state $t_0 \in U$. Since U is an EC, $\vec{\nu}_U \in \mathsf{ExpSol}^+_{\mathcal{G} \downharpoonleft U}(t, h_U)$ for *every* state $t \in U$. Since $\vec{\nu}_U > \vec{0}$, we can apply Lemma 3 for every $t \in U$, and obtain a strategy h_t s.t. $(\vec{0}; \vec{\nu}_U) \in \mathsf{BWCSol}^+_{\mathcal{G} \downharpoonleft U}(t, h_t)$. Let h_U be the finite-memory strategy in $\mathcal{G} \downharpoonleft U$ that plays according to h_t when starting from state t. Clearly, $(\vec{0}; \vec{\nu}_U) \in \mathsf{BWCSol}^+_{\mathcal{G} \downharpoonleft U}(t, h_U)$.

Consider the strategy h^{cmb}_N parameterized by a natural number $N > 0$ which is defined as follows:

1) Play according to $\hat{h}$ for N steps.
2) After N steps:
    2a) If we are inside the MWEC U, then switch to h_U.
    2b) Otherwise, play according to h^{wc}.

We argue that h^{cmb}_N satisfies the beyond worst-case objective $(\vec{0}; \vec{\nu}) \in \mathsf{BWCSol}^+_{\mathcal{G}}(s_0, h^{cmb}_N)$ for N large enough. For every N, h^{cmb}_N clearly satisfies the worst-case objective, since after

N steps it switches to a strategy that satisfies it by construction (by prefix-independence of the mean payoff objective). We now consider the expectation objective. By Point 1 of Proposition 4, the set of states visited infinitely often by $\hat{h}$ is a subset of the MWEC U with probability y^*_U. By taking N large enough, we can guarantee being inside U with probability arbitrarily close to y^*_U. By construction, h_U can be chosen to achieve expected mean payoff arbitrarily close to $\vec{\nu}_U$. Since h^{cmb}_N switches to h_U with probability arbitrarily close to y^*_U, h^{cmb}_N achieves expected mean payoff arbitrarily close to $\sum_{\text{MWEC } U} y^*_U \cdot \vec{\nu}_U$. By Point 3 of Proposition 4, the latter quantity is $> \vec{\nu}$. There exists N^* large enough s.t. $h^{cmb}_{N^*}$ achieves expected mean payoff $> \vec{\nu}$. Take $h^{cmb} = h^{cmb}_{N^*}$. As required, $\vec{\nu} \in \mathsf{ExpSol}^+_{\mathcal{G}}(s_0, h^{cmb})$. □

Running example. Since $U = \{t\}$ is a MWEC, while $V = \{u, v\}$ is not, finite memory strategies must go to U. Therefore, with finite memory we can ensure BWC threshold $((0,0);(0,9))$, but not $((0,0);(9,9))$ for example.

3) Complexity: We obtain the following complexity characterization for the threshold problem with finite-memory controllers.

Theorem 5. *The finite-memory multidimensional mean-payoff BWC threshold problem is* coNP*-complete.*

Proof. Pruning states where the worst-case objective cannot be satisfied requires solving multidimensional mean-payoff games, which can be done in coNP by Theorem 2. It has been already shown in [3] how the decomposition in MWEC can be performed in $\mathsf{P}$with an oracle for solving mean-payoff games. Thus, the MWEC decomposition can be performed in coNP. System T has size polynomial in $\mathcal{G}$ (there are only polynomially many MWECs) and it can thus be produced in coNP. By Lemma 4, it suffices to solve system T, which can be done in polynomial time by linear programming. The lower bound follows directly from the fact that the multidimensional BWC threshold problem contains the worst-case as a subproblem; the latter is coNP-hard as recalled in Theorem 2. □

The complexity of the BWC problem is dominated by the worst-case subproblem. We obtain an improved complexity by restricting the worst-case to be essentially unidimensional. Formally, we say that a BWC threshold $(\vec{\mu}; \vec{\nu}) \in \mathbb{R}^{2d}$ has *trivial worst-case component* i, with $1 \leq i \leq d$, iff $\vec{\mu}[i] = -W$, where W is the maximal absolute value of any weight in $\mathcal{G}$. We say that $(\vec{\mu}; \vec{\nu})$ *is essentially worst-case unidimensional* iff it has at most one non-trivial worst-case component. We can ignore trivial components when solving a worst-case threshold problem. Thus, the worst-case problem for essentially unidimensional thresholds reduces to a simple unidimensional worst-case problem. As recalled in Theorem 3, the latter can be solved in $\mathsf{NP}\cap\mathsf{coNP}$, thus yielding the following improved complexity for the BWC problem.

Corollary 1. *The finite-memory multidimensional mean-payoff BWC threshold problem w.r.t.* essentially worst-case unidimensional thresholds *is in* $\mathsf{NP}\cap\mathsf{coNP}$.

Since the *unidimensional* BWC problem, i.e., where all weights are unidimensional, is in $\mathsf{NP}\cap\mathsf{coNP}$ (cf. Theorem 4), this results shows that we can add a multidimensional expectation objective to a unidimensional worst-case obligation without an extra price in complexity. In particular, we can model complex situations like the task system presented in Sec. I-C, where the worst-case and expectation mean payoffs are along independent dimensions.

B. Infinite-memory synthesis

Already in the unidimensional case, infinite-memory strategies are more powerful than finite-memory ones (cf. [4, Fig. 6]). This is a consequence of the fact that finite-memory strategies for the BWC objective ultimately remain inside WECs almost surely (cf. Proposition 2). On the other hand, infinite-memory strategies can benefit from payoffs achievable inside arbitrary ECs. In this section, we address the problem of deciding whether there exists a general strategy, i.e., not necessarily finite-memory one, for the multidimensional BWC problem. This was left as an open problem, already in the unidimensional case [3]. As in the previous section, we first analyze ECs, and then general MDPs.

1) Inside an EC: The lemma below is a direct generalization of Lemma 2 to arbitrary ECs. While for WECs we could construct finite-memory strategies, we now construct infinite-memory strategies for arbitrary ECs.

Lemma 5. *Let $\mathcal{G}$ be a pruned multidimensional mean-payoff MDP, let s_0 be a state in an* EC *U of MDP, and let $\vec{\nu} \geq \vec{0}$ be an expectation vector $\vec{\nu} \in \mathsf{ExpSol}^+_{\mathcal{G}\restriction U}(s_0)$ which is achievable while remaining in U. There exists a strategy $f \in \Delta(\mathcal{G})$ (not necessarily remaining in U) s.t. $(\vec{0}; \vec{\nu}) \in \mathsf{BWCSol}^+_{\mathcal{G}}(s_0, f)$.*

Remark 5. The statement of the lemma holds even with f a *pure* strategy, by applying Remark 1 when constructing the expectation strategy f^{exp} below. However, randomized strategies suffice for our purposes.

The rest of this section is devoted to the proof of Lemma 5. We proceed by combining in a non-trivial way a strategy for the expectation with a strategy for the worst-case. Let f^{wc} be a worst-case strategy s.t. $\vec{0} \in \mathsf{WCSol}^+_{\mathcal{G}}(s, f^{wc})$ for every state s, which exists since the $\mathcal{G}$ is pruned. Let f^{exp} be a expectation strategy s.t. $\vec{\nu} \in \mathsf{ExpSol}^+_{\mathcal{G}\restriction U}(s_0, f^{exp})$. By Lemma 1, we can assume that f^{exp} is finite-memory and unichain. For technical reasons, it is convenient to assume that f^{exp} is finite-memory, even though we are going to construct a infinite-memory strategy. Moreover, since we are in an EC, we can further assume that f^{exp} achieves expectation $> \vec{\nu}$ from *every* state of the EC.

The idea is to play according to two different modes. In the first mode, we play according to f^{exp}, and in the second mode according to f^{wc}. We start in the first mode, and possibly go to the second mode according to certain conditions. This happens with a certain probability, which we call *switching probability*. Once in the second mode, we remain in the second mode. In order to achieve an expectation arbitrarily close to that achieved by f^{exp}, we need to be able to make the switching probability arbitrarily small. At the same time, in order to ensure that the worst-case objective is satisfied, we need to guarantee that, when no switch occurs, the mean payoff is surely $> \vec{0}$. (If a switch occurs, the worst-case is satisfied by the definition of f^{wc}.) These two constraints are conflicting and make the construction of a combined strategy non-trivial.

The *combined* strategy f_K is parameterized by a natural number $K > 0$. In order to decide whether to switch to the second mode or not, we keep track of the total payoff since the beginning of the play as a vector in $\mathbb{Z}^d$. This value is unbounded in general, and this is explains why the strategy uses infinite memory. Let $\vec{N}_i$ be

$$\vec{N}_i = \frac{\vec{\nu} \cdot i \cdot K}{2}.$$

Thus, during the first mode the expected total payoff at the end of phase i is $> 2 \cdot \vec{N}_{i+1}$. The first mode is split into *phases*, each of length K. During phase $i \geq 0$, we play according to f^{exp} for at most K steps. There are two conditions that can trigger a switch to the second mode:

[Switching condition 1 (SC1)] If we are in phase $i \geq 1$ and the total payoff since the beginning of the play is not always $> \vec{N}_i$ during the current phase, then switch to f^{wc} permanently.

[Switching condition 2 (SC2)] If the total payoff since the beginning of the play is not $> 2 \cdot \vec{N}_{i+1}$ at the end of the current phase, then switch to f^{wc} permanently.

What it remains to do is to show that we can choose $K > 0$ in order to satisfy the BWC objective. First, we show that, for every choice of the parameter K, the combined strategy f_K guarantees the worst-case objective.

Proposition 5. *For every $K \in \mathbb{N}$ and state s_0 in the EC U, $\vec{0} \in \mathsf{WCSol}^+_{\mathcal{G}}(s_0, f_K)$.*

Proof. There are two cases to consider. If we ever switch to the second mode, then the run is eventually consistent with the worst-case strategy f^{wc}, which guarantees worst-case mean payoff $> \vec{0}$ (by prefix independence). Otherwise, assume that we never leave the first mode. During phase $i \geq 1$ the total payoff is always $> \vec{N}_i = \frac{\vec{\nu} \cdot i \cdot K}{2}$, and the total length of the play is at most $i \cdot K$. The average mean payoff during phase i is uniformly $> \frac{\vec{\nu}}{2}$. The limit inferior of the average mean payoff is also $> \frac{\vec{\nu}}{2} \geq \vec{0}$. □

We conclude by showing that K can be chosen s.t. the combined strategy f_K achieves expected mean payoff $> \vec{\nu}$.

Lemma 6. *There exists $K \in \mathbb{N}$ s.t. $\vec{\nu} \in \mathsf{ExpSol}^+_{\mathcal{G}}(s_0, f_K)$.*

Proof. We show that we can choose a $K > 0$ large enough s.t. the switching probability is negligible, and thus the impact of switching to the worst-case strategy f^{wc} on the expected mean payoff is also negligible. For now fix an arbitrary $K > 0$, and consider the Markov chain $\mathcal{G}[f_K]$. Let p_K be the probability to switch to the second mode due to SC1 in any phase $i \geq 1$, and let q_K be the probability to switch to the second mode due to SC2 in any phase $i \geq 0$. Thus, with probability at most $1 - (1 - p_K) \cdot (1 - q_K)$ we switch to the second mode. By prefix independence of the mean payoff objective, the expected mean payoff achieved by f_K satisfies:

$$\begin{aligned}\mathbb{E}^{\mathcal{G}}_{s_0, f_K}[\mathsf{MP}] \geq \quad & (1 - (1 - p_K) \cdot (1 - q_K)) \cdot \mathbb{E}^{\mathcal{G}}_{s_0, f^{wc}}[\mathsf{MP}] + \\ & + (1 - p_K) \cdot (1 - q_K) \cdot \mathbb{E}^{\mathcal{G}}_{s_0, f^{exp}}[\mathsf{MP}]\end{aligned}$$

Since $\mathbb{E}^{\mathcal{G} \downharpoonright U}_{s, f^{exp}}[\mathsf{MP}] > \vec{\nu}$ by definition, it suffices to show that both probabilities p_K and q_K can be made arbitrarily small. We argue about them separately.

Let $p_{i,K}$ be the probability of switching to the second mode due to SC1 during phase $i \geq 1$, i.e, the probability that the total payoff goes below $\vec{N}_i$ in any component:

$$p_{i,K} = \mathbb{P}^{\mathcal{G}[f_K]}_{s_0}\left[\exists (K \cdot i \leq h < K \cdot (i+1)) \cdot \mathsf{TP}_h \not> \vec{N}_i\right]$$

Then, $p_K = p_{1,K} + (1 - p_{1,K}) \cdot p_{2,K} + (1 - p_{1,K}) \cdot (1 - p_{2,K}) \cdot p_{3,K} + \ldots$, and thus $p_K \leq p_{1,K} + p_{2,K} + \ldots$. We claim the following exponential upper bound on $p_{i,K}$.

Claim 1. *There are rational constants a and b with $b < 1$ s.t., for every $i \geq 1$ and for sufficiently large K, $p_{i,K} \leq a \cdot b^{K \cdot i}$. Note that a and b do not depend neither on K, nor on i.*

By the claim, $p_K \leq a \cdot (b^K + b^{K \cdot 2} + \ldots) \leq a \cdot b^K / (1 - b^K)$, and thus $\lim_K p_K = 0$ since $b^K < 1$.

Let $q_{i,K}$ be the probability of switching to the second mode due to SC2 at the end of phase $i \geq 0$. Thus, $q_{i,K}$ is the probability that, at the end of phase i, the total payoff is less than $2 \cdot \vec{N}_{i+1} = \vec{\nu} \cdot (i+1) \cdot K$ in any component:

$$q_{i,K} = \mathbb{P}^{\mathcal{G}[f_K]}_{s_0}\left[\mathsf{TP}_{K \cdot (i+1)} \not> 2 \cdot \vec{N}_{i+1}\right]$$

We have $q_K = q_{0,K} + (1 - q_{0,K}) \cdot q_{1,K} + (1 - q_{0,K}) \cdot (1 - q_{1,K}) \cdot q_{2,K} + \ldots$. We show $\lim_K q_K = 0$ as in the last paragraph, by the following claim.

Claim 2. *There exist rational constants a and b with $b < 1$ s.t., for every $i \geq 0$ and sufficiently large K, $q_{i,K} \leq a \cdot b^{K \cdot (i+1)}$. Note that a and b do not depend neither on K, nor on i.*

Both claims are proved in the technical report [12]. □

2) The general case: As in the synthesis for finite-memory strategies (cf. Sec. III-A), we reduce the infinite-memory BWC problem to the solution of a system of linear inequalities. The new system of equations T' is shown in Fig. 5. It is obtained as a modification of system T from the finite-memory case shown in Fig. 4: Specifically, T' is the same as T, except that Equations (A2), (B), and (C3) are interpreted w.r.t. MEC (instead of MWEC). The correctness of the reduction is stated in the lemma below.

Lemma 7. *Let $\mathcal{G}$ be a pruned multidimensional mean-payoff MDP, let $\vec{\nu} \geq \vec{0}$, and let $s_0 \in S$. There exists a (possibly infinite-memory) strategy h s.t. $(\vec{0}; \vec{\nu}) \in \mathsf{BWCSol}^+_{\mathcal{G}}(s_0, h)$, if, and only if, the system T' has a non-negative solution.*

Proof sketch. The proof is analogous to the proof of Lemma 4. The crucial difference is that, by the modifications performed to obtain T' from T, we obtain strategies which almost surely

$$1_{s_0}(s) + \sum_{(r,s)\in E} y_{rs} = \sum_{(s,t)\in E} y_{st} + y_s \qquad \forall s \in S \tag{A1}$$

$$y_{st} = R(s)(t) \cdot (\sum_{(r,s)\in E} y_{rs} - y_s) \qquad \forall (s,t) \in E \text{ with } s \in S^R \tag{A1'}$$

$$\sum_{\text{MEC } U} \sum_{s\in U} y_s = 1 \tag{A2-bis}$$

$$\sum_{s\in U} y_s = \sum_{(r,s)\in E\cap U\times U} x_{rs} \qquad \forall \text{ MEC } U \tag{B-bis}$$

$$\sum_{(r,s)\in E} x_{rs} = \sum_{(s,t)\in E} x_{st} \qquad \forall s \in S \tag{C1}$$

$$x_{st} = R(s)(t) \cdot \sum_{(r,s)\in E} x_{rs} \qquad \forall (s,t) \in E \text{ with } s \in S^R \tag{C1'}$$

$$\sum_{(s,t)\in E} x_{st} \cdot w(s,t)[i] > \vec{\nu}[i] \qquad \forall (1 \le i \le n) \tag{C2}$$

$$\sum_{(s,t)\in E\cap U\times U} x_{st} \cdot w(s,t)[i] > 0 \qquad \forall \text{ MEC } U, 1 \le i \le n \tag{C3-bis}$$

Fig. 5: Linear system T' for the BWC infinite-memory threshold problem.

stay forever inside ECs, instead of WECs. Since we are allowed infinite-memory, we can approximate the BWC objective inside ECs by replacing Lemma 2 with Lemma 5. □

Running example. An infinite-memory strategy can benefit both from the expectation $(5, 15)$ in U and from $(15, 5)$ in V (which is not a WEC). By going to either EC with equal probability and playing according to a local BWC strategy, an infinite-memory strategy can ensure, for every $\varepsilon > 0$, the BWC threshold $((0, 0); (10 - \varepsilon, 10 - \varepsilon))$.

3) Complexity: We obtain the following complexity result for the threshold problem for arbitrary controllers.

Theorem 6. *The multidimensional mean-payoff BWC threshold problem is* coNP*-complete.*

Proof. Pruning the game to remove states which are losing for the worst-case objective requires solving a multidimensional mean-payoff game, which is coNP-complete by Theorem 2. Then, by Lemma 7, it suffices to solve system L'. Notice that system L' is of polynomial size since there are only polynomially many *maximal* ECs. □

Again, the complexity of the BWC problem is dominated by the worst-case. By restricting to essentially worst-case unidimensional thresholds we obtain a better complexity.

Corollary 2. *The multidimensional mean-payoff BWC threshold problem w.r.t.* essentially worst-case unidimensional thresholds *is in* NP∩coNP.

This solves with optimal complexity the infinite-memory unidimensional BWC problem, which was left open in [3].

IV. BEYOND ALMOST-SURE SYNTHESIS

We introduce a natural relaxation of the BWC problem which enjoys a better complexity. Intuitively, we replace the worst-case objective in the BWC problem with a weaker almost sure objective. While the BWC problem is coNP-complete, we show that this relaxation can be solved in P, even in the multidimensional setting. A similar result has recently been obtained in [5]. Given an MDP $\mathcal{G}$, a starting state s_0 therein, and a Controller's strategy $f \in \Delta(\mathcal{G})$, the set of *almost sure achievable solutions for* f, denoted $\mathsf{ASSol}^+{}_{\mathcal{G}}(s_0, f)$, is the set of vectors $\vec{\mu} \in \mathbb{R}^d$ s.t. Controller can almost surely guarantee mean payoff $> \vec{\mu}$ when playing according to f, i.e., $\mathsf{ASSol}^+{}_{\mathcal{G}}(s, f) = \left\{ \vec{\mu} \in \mathbb{R}^d \;\middle|\; \mathbb{P}^{\mathcal{G}}_{s,f}[\mathsf{MP} > \vec{\mu}] = 1 \right\}$. The set of *beyond almost-sure achievable solutions for* f, denoted $\mathsf{BASSol}^+_{\mathcal{G}}(s_0, f)$, is the set of pairs of vectors $(\vec{\mu}; \vec{\nu}) \in \mathbb{R}^{2d}$ s.t. Controller can almost surely guarantee mean payoff $> \vec{\mu}$ and achieve expected mean payoff $> \vec{\nu}$ when starting from s_0 and playing according to f, i.e.,

$$\mathsf{BASSol}^+_{\mathcal{G}}(s_0, f) = \left\{ (\vec{\mu}; \vec{\nu}) \in \mathbb{R}^{2d} \;\middle|\; \begin{array}{c} \vec{\mu} \in \mathsf{ASSol}^+{}_{\mathcal{G}}(s_0, f) \\ \text{and} \\ \vec{\nu} \in \mathsf{ExpSol}^+_{\mathcal{G}}(s_0, f) \end{array} \right\}$$

The set of *beyond almost-sure achievable solutions* is $\mathsf{BASSol}^+_{\mathcal{G}}(s_0) = \bigcup_{f\in\Delta(\mathcal{G})} \mathsf{BASSol}^+_{\mathcal{G}}(s_0, f)$. Given $(\vec{\mu}; \vec{\nu}) \in \mathbb{R}^{2d}$ and a state s_0, the *beyond almost-sure threshold problem* asks whether $(\vec{\mu}; \vec{\nu}) \in \mathsf{BASSol}^+_{\mathcal{G}}(s_0)$.

Remark 6. We assume w.l.o.g. that $\vec{\mu} = \vec{0}$ and $\vec{\nu} \ge \vec{0}$. The first condition is ensured by subtracting $\vec{\mu}$ everywhere. The second condition follows from the observation that, if the mean payoff is $> \vec{0}$ almost surely, then also the expectation is $> \vec{0}$ surely.

We observe that, inside an EC, there is no trade-off between the almost sure and the expectation objective.

Lemma 8. *Let $\mathcal{G}$ be a multidimensional mean-payoff MDP, let s_0 be a state in an EC U thereof, and let $\vec{\nu} \in \mathsf{ExpSol}^{+}_{\mathcal{G}\downharpoonright U}(s_0)$ be an expectation achievable while remaining inside U. There exists a* finite-memory *strategy $g \in \Delta_F(\mathcal{G})$ s.t. $(\vec{\nu};\vec{\nu}) \in \mathsf{BASSol}^{+}_{\mathcal{G}\downharpoonright U}(s_0, g)$ which also remains inside U.*

Proof. By Lemma 1, there exists a finite-memory strategy g s.t. $\vec{\nu} \in \mathsf{ExpSol}^{+}_{\mathcal{G}\downharpoonright U}(s_0, g)$ and $\mathcal{G}[g]$ is unichain. Consequently, the mean payoff is $> \vec{\nu}$ almost surely. □

Thus, most of the effort goes in analyzing the general case. As in the BWC problem, we reduce the BAS problem to the solution of a system of linear inequalities. We assume that from state s_0 all ECs are reachable with positive probability. It turns out that the same system of equations T' used in the infinite-memory BWC threshold problem also solves the BAS problem. We obtain a better complexity since we do not require the MDP to be pruned (which avoids solving an expensive mean-payoff game).

Theorem 7. *The multidimensional mean-payoff BAS threshold problem is in* P.

Proof. The proof of correctness is the same as in Lemma 7, where Lemma 8 replaces Lemma 5 in the analysis of ECs. Crucially for complexity, we do not need to assume that the MDP is pruned. Therefore, system T' can be built (and solved) in P. Since Lemma 8 even yields finite-memory strategies inside an EC, the construction of Lemma 7 shows that finite-memory strategies suffice for the BAS threshold problem. (This relies on the strict BAS semantics. If non-strict inequalities are used, then the problem can still be solved in P but the construction above yields an infinite-memory strategy, and infinite-memory strategies are more powerful than finite-memory ones for the non-strict BAS problem; cf. also [5].) □

Running example. The BAS problem is strictly weaker than the BWC problem. Consider the MDP from Fig. 3 without the edge (u, t). This modification makes both states u and v losing for the worst-case, thus they are pruned away when solving the BWC problem (even with infinite memory). On the other hand, the mean payoff is almost surely $(5, 15)$ from V, and thus it satisfies the almost sure objective $> (0, 0)$. Therefore, for every $\varepsilon > 0$, we can achieve the BAS threshold $((0, 0); (10-\varepsilon, 10-\varepsilon))$ by going to t or u with equal probability.

V. Conclusions

In this paper, we studied the multidimensional generalization of the beyond worst-case problem introduced by Bruyère *et al.* [3]. We have provided tight coNP-completeness results for the this problem under both finite-memory and general strategies. Since multidimensional mean-payoff games are already coNP-complete, our upper bound shows that we can add a multidimensional expectation optimization objective on top of a worst-case requirement without a corresponding increase in complexity. Notice that, while infinite-memory strategies were known to be more powerful than finite-memory ones already in the unidimensional setting [3], the corresponding synthesis problem was left open. Our results thus complete the complexity picture for this problem. Moreover, we showed that, when the worst-case objective is unidimensional, the complexity reduces to $\mathsf{NP}\cap\mathsf{coNP}$, and this holds even for multidimensional expectations. This generalizes with optimal complexity the $\mathsf{NP}\cap\mathsf{coNP}$ upper bound for the unidimensional beyond worst-case problem [3]. From a practical point of view, our reductions to linear programming can be performed in pseudo-polynomial time by using the results of [16] for unidimensional mean-payoff games, and [17] for *fixed* number of dimensions. Furthermore, we introduced the beyond almost-sure problem as a natural relaxation of the beyond worst-case problem, by weakening the worst-case requirement to an almost-sure one. This natural relaxation enjoys a polynomial time solution and finite memory strategies always suffice. Moreover, our reduction to linear programming shows that the beyond almost-sure problem is amenable to be solved efficiently in practice, and thus it has the strongest appeal for practical applications.

References

[1] T. M. Liggett and S. A. Lippman, "Stochastic games with perfect information and time average payoff," *SIAM Review*, vol. 11, no. 4, pp. 604–607, 1969. [Online]. Available: http://www.jstor.org/stable/2029090

[2] A. Ehrenfeucht and J. Mycielski, "Positional strategies for mean payoff games," *International Journal of Game Theory*, vol. 8, no. 2, pp. 109–113, 1979. [Online]. Available: http://dx.doi.org/10.1007/BF01768705

[3] V. Bruyère, E. Filiot, M. Randour, and J.-F. Raskin, "Meet Your Expectations With Guarantees: Beyond Worst-Case Synthesis in Quantitative Games," in *Proc. of STACS'14*, ser. LIPIcs, vol. 25, 2014, pp. 199–213.

[4] ——, "Meet your expectations with guarantees: Beyond worst-case synthesis in quantitative games," Tech. Rep., 09 2013. [Online]. Available: http://arxiv.org/abs/1309.5439

[5] K. Chatterjee, Z. Komárková, and J. Křetínský, "Unifying two views on multiple mean-payoff objectives in markov decision processes," *To appear in Proc. of LICS'15*.

[6] M. Puterman, *Markov Decision Processes - Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[7] Y. Velner and A. Rabinovich, "Church synthesis problem for noisy input," in *Proc. of FOSSACS'11*. Springer, 2011, pp. 275–289. [Online]. Available: http://dl.acm.org/citation.cfm?id=1987171.1987194

[8] Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. Rabinovich, and J.-F. Raskin, "The complexity of multi-mean-payoff and multi-energy games," Tech. Rep., 09 2012. [Online]. Available: http://arxiv.org/abs/1209.3234

[9] T. Brázdil, V. Brozek, K. Chatterjee, V. Forejt, and A. Kucera, "Two views on multiple mean-payoff objectives in markov decision processes," *LMCS*, vol. 10, no. 1, 2014. [Online]. Available: http://dx.doi.org/10.2168/LMCS-10(1:13)2014

[10] M. Randour, J.-F. Raskin, and O. Sankur, "Percentile queries in multi-dimensional markov decision processes," in *Proc. of CAV'15*, to appear.

[11] U. Zwick and M. Paterson, "The complexity of mean payoff games on graphs," *TCS*, vol. 158, no. 1–2, pp. 343–359, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0304397595001883

[12] L. Clemente and J.-F. Raskin, "Multidimensional beyond worst-case and almost-sure problems for mean-payoff objectives," 04 2015. [Online]. Available: http://arxiv.org/abs/1504.08211

[13] P. Billingsley, *Probability and Measure*. John Wiley & Sons, 1995.

[14] L. De Alfaro, "Formal verification of probabilistic systems," Ph.D. dissertation, Stanford University, Stanford, CA, USA, 1998.

[15] M. Jurdziński, "Deciding the winner in parity games is in UP∩co-UP," *Information Processing Letters*, vol. 68, no. 3, pp. 119–124, 1998.

[16] L. Brim, J. Chaloupka, L. Doyen, R. Gentilini, and J. Raskin, "Faster algorithms for mean-payoff games," *Formal Methods in System Design*, vol. 38, no. 2, pp. 97–118, 2011.

[17] K. Chatterjee and Y. Velner, "Hyperplane separation technique for multidimensional mean-payoff games," in *Proc. of CONCUR'13*, ser. LNCS, vol. 8052. Springer, pp. 500–515.