Conditional Value-at-Risk for Reachability and Mean Payoff in Markov Decision Processes

Jan Křetínský
Institut für Informatik (I7)
Technische Universität München
Garching bei München, Bavaria, Germany
jan.kretinsky@in.tum.de

Tobias Meggendorfer
Institut für Informatik (I7)
Technische Universität München
Garching bei München, Bavaria, Germany
tobias.meggendorfer@in.tum.de

Abstract

We present the *conditional value-at-risk (CVaR)* in the context of Markov chains and Markov decision processes with reachability and mean-payoff objectives. CVaR quantifies risk by means of the expectation of the worst p-quantile. As such it can be used to design risk-averse systems. We consider not only CVaR constraints, but also introduce their conjunction with expectation constraints and quantile constraints (value-at-risk, VaR). We derive lower and upper bounds on the computational complexity of the respective decision problems and characterize the structure of the strategies in terms of memory and randomization.

CCS Concepts • **Theory of computation → Verification by model checking**;

ACM Reference Format:
Jan Křetínský and Tobias Meggendorfer. 2018. Conditional Value-at-Risk for Reachability and Mean Payoff in Markov Decision Processes. In *LICS '18: 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, July 9–12, 2018, Oxford, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3209108.3209176

1 Introduction

Markov decision processes (MDP) are a standard formalism for modelling stochastic systems featuring non-determinism. The fundamental problem is to design a strategy resolving the non-deterministic choices so that the systems' behaviour is optimized with respect to a given objective function, or, in the case of multi-objective optimization, to obtain the desired trade-off. The objective function (in the optimization phrasing) or the query (in the decision-problem phrasing) consists of two parts. First, a payoff is a measurable function assigning an outcome to each run of the system. It can be real-valued, such as the *long-run average reward* (also called *mean payoff*), or a two-valued predicate, such as *reachability*. Second, the payoffs for single runs are combined into an overall outcome of the strategy, typically in terms of *expectation*. The resulting objective function is then for instance the expected long-run average reward, or the probability to reach a given target state.

Risk-averse control aims to overcome one of the main disadvantages of the expectation operator, namely its ignorance towards the incurred risks, intuitively phrased as a question *"How bad are the bad cases?"* While the standard deviation (or variance) quantifies the spread of the distribution, it does not focus on the bad cases and thus fails to capture the risk. There are a number of quantities used to deal with this issue:

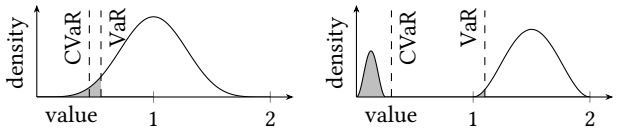

Figure 1. Illustration of VaR and CVaR for some random variables.

- The *worst-case* analysis (in the financial context known as discounted maximum loss) looks at the payoff of the worst possible run. While this makes sense in a fully non-deterministic environment and lies at the heart of verification, in the probabilistic setting it is typically unreasonably pessimistic, taking into account events happening with probability 0, e.g., never tossing head on a fair coin.
- The *value-at-risk* (VaR) denotes the worst p-quantile for some $p \in [0, 1]$. For instance, the value at the 0.5-quantile is the median, the 0.05-quantile (the *vigintile* or *ventile*) is the value of the best run among the 5% worst ones. As such it captures the "reasonably possible" worst-case. See Fig. 1 for an example of VaR for two given probability density functions. There has been an extensive effort spent recently on the analysis of MDP with respect to VaR and the re-formulated notions of quantiles, percentiles, thresholds, satisfaction view etc., see below. Although VaR is more realistic, it tends to ignore outliers too much, as seen in Fig. 1 on the right. VaR has been characterized as *"seductive, but dangerous"* and *"not sufficient to control risk"* [8].
- The *conditional value-at-risk* (average value-at-risk, expected shortfall, expected tail loss) answers the question *"What to expect in the bad cases?"* It is defined as the expectation over all events worse than the value-at-risk, see Fig. 1. As such it describes the lossy tail, taking outliers into account, weighted respectively. In the degenerate cases, CVaR for $p = 1$ is the expectation and for $p = 0$ the (probabilistic) worst case. It is an established risk metric in finance, optimization and operations research, e.g. [1, 33], and *"is considered to be a more consistent measure of risk"* [33]. Recently, it started permeating to areas closer to verification, e.g. robotics [13].

Our contribution In this paper, we investigate optimization of MDP with respect to CVaR as well as the respective trade-offs with expectation and VaR. We study the VaR and CVaR operators for the first time with the payoff functions of weighted reachability and

LICS '18, July 9–12, 2018, Oxford, United Kingdom

ACM ISBN 978-1-4503-5583-4/18/07…$15.00
https://doi.org/10.1145/3209108.3209176

mean payoff, which are fundamental in verification. Moreover, we cover both the single-dimensional and the multi-dimensional case.

Particularly, we define CVaR for MDP and show the peculiarities of the concept. Then we study the computational complexity and the strategy complexity for various settings, proving the following:

- The single dimensional case can be solved in polynomial time through linear programming, see Section 5.
- The multi-dimensional case is NP-hard, even for CVaR-only constraints. Weighted reachability is NP-complete and we give PSPACE and EXPSPACE upper bounds for mean payoff with CVaR and expectation constraints, and with additional VaR constraints, respectively, see Section 6. (Note that already for the sole VaR constraints only an exponential algorithm is known; the complexity is an open question and not even NP-hardness is known [15, 32].)
- We characterize the strategy requirements, both in terms of memory, ranging from memoryless, over constant-size to infinite memory, and the required degree of randomization, ranging from fully deterministic strategies to randomizing strategies with stochastic memory update.

While dealing with the CVaR operator, we encountered surprising behaviour, preventing us to trivially adapt the solutions to the expectation and VaR problems:

- Compared to, e.g., expectation and VaR, CVaR does not behave linearly w.r.t. stochastic combination of strategies.
- A conjunction of CVaR constraints already is NP-hard, since it can force a strategy to play deterministically.

1.1 Related work

Worst case Risk-averse approaches optimizing the worst case together with expectation have been considered in beyond-worst-case and beyond-almost-sure analysis investigated in both the single-dimensional [11] and in the multi-dimensional [17] setup.

Quantiles The decision problem related to VaR has been phrased in probabilistic verification mostly in the form *"Is the probability that the payoff is higher than a given value threshold more than a given probability threshold?"* The total reward gained attention both in the verification community [6, 24, 35] and recently in the AI community [23, 29]. Multi-dimensional percentile queries are considered for various objectives, such as mean-payoff, limsup, liminf, shortest path in [32]; for the specifics of two-dimensional case and their interplay, see [3]. Quantile queries for more complex constraints have also been considered, namely their conjunctions [9, 20], conjunctions with expectations [15] or generally Boolean expressions [25]. Some of these approaches have already been practically applied and found useful by domain experts [4, 5].

CVaR There is a body of work that optimizes CVaR in MDP. However, to the best of our knowledge, all the approaches (1) focus on the single-dimensional case, (2) disregard the expectation, and (3) treat neither reachability nor mean payoff. They focus on the discounted [7], total [13], or immediate [27] reward, as well as extend the results to continuous-time models [26, 30]. This work comes from the area of optimization and operations research, with the notable exception of [13], which focuses on the total reward. Since the total reward generalizes weighted reachability, [13] is related to our work the most. However, it provides only an approximation solution for the one-dimensional case, neglecting expectation and the respective trade-offs.

Further, CVaR is a topic of high interest in finance, e.g., [8, 33]. The central difference is that there variations of portfolios (i.e. the objective functions) are considered while leaving the underlying random process (the market) unchanged. This is dual to our problem, since we fix the objective function and now search for an optimal random process (or the respective strategy).

Multi-objective expectation In the last decade, MDP have been extensively studied generally in the setting of multiple objectives, which provides some of the necessary tools for our trade-off analysis. Multiple objectives have been considered for both qualitative payoffs, such as reachability and LTL [19], as well as quantitative payoffs, such as mean payoff [9], discounted sum [14], or total reward [22]. Variance has been introduced to the landscape in [10].

2 Preliminaries

Due to space constraints, some proofs and explanations are shortened or omitted when clear and can be found in [28].

2.1 Basic definitions

We mostly follow the definitions of [9, 15]. $\mathbb{N}, \mathbb{Q}, \mathbb{R}$ are used to denote the sets of positive integers, rational and real numbers, respectively. For $n \in \mathbb{N}$, let $[n] = \{1, \dots, n\}$. Further, k_j refers to $k \cdot e_j$, where e_j is the unit vector in dimension j.

We assume familiarity with basic notions of probability theory, e.g., *probability space* $(\Omega, \mathcal{F}, \mu)$, *random variable* F, or *expected value* $\mathbb{E}$. The set of all distributions over a countable set C is denoted by $\mathcal{D}(C)$. Further, $d \in \mathcal{D}(C)$ is Dirac if $d(c) = 1$ for some $c \in C$. To ease notation, for functions yielding a distribution over some set C, we may write $f(\cdot, c)$ instead of $f(\cdot)(c)$ for $c \in C$.

Markov chains A *Markov chain* (MC) is a tuple $\mathsf{M} = (S, \delta, \mu_0)$, where S is a countable set of states[1], $\delta : S \to \mathcal{D}(S)$ is a probabilistic transition function, and $\mu_0 \in \mathcal{D}(S)$ is the initial probability distribution. The SCCs and BSCCs of a MC are denoted by SCC and BSCC, respectively [31].

A *run* in M is an infinite sequence $\rho = s_1 s_2 \cdots$ of states, we write ρ_i to refer to the i-th state s_i. A *path* ϱ in M is a finite prefix of a run ρ. Each path ϱ in M determines the set $\mathsf{Cone}(\varrho)$ consisting of all runs that start with ϱ. To M, we associate the usual probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the set of all runs in M, $\mathcal{F}$ is the σ-field generated by all $\mathsf{Cone}(\varrho)$, and $\mathbb{P}$ is the unique probability measure such that $\mathbb{P}(\mathsf{Cone}(s_1 \cdots s_k)) = \mu_0(s_1) \cdot \prod_{i=1}^{k-1} \delta(s_i, s_{i+1})$. Furthermore, $\Diamond B$ ($\Diamond\Box B$) denotes the set of runs which eventually reach (eventually remain in) the set $B \subseteq S$, i.e. all runs where $\rho_i \in B$ for some i (there exists an i_0 such that $\rho_i \in B$ for all $i \geq i_0$).

Markov decision processes A *Markov decision process* (MDP) is a tuple $\mathcal{M} = (S, \mathsf{A}, \mathsf{Av}, \Delta, s_0)$ where S is a finite set of states, A is a finite set of actions, $\mathsf{Av} : S \to 2^{\mathsf{A}} \setminus \{\emptyset\}$ assigns to each state s the set $\mathsf{Av}(s)$ of actions enabled in s so that $\{\mathsf{Av}(s) \mid s \in S\}$ is a partitioning of A[2], $\Delta : \mathsf{A} \to \mathcal{D}(S)$ is a probabilistic transition function that given an action a yields a probability distribution over the successor states, and s_0 is the initial state of the system.

[1] We allow the state set to be countable for the formal definition of strategies on MDP. When dealing with Markov Chains in queries, we only consider finite state sets.
[2] In other words, each action is associated with exactly one state.

A *run* ρ of $\mathcal{M}$ is an infinite alternating sequence of states and actions $\rho = s_1 a_1 s_2 a_2 \cdots$ such that for all $i \geq 1$, we have $a_i \in \mathsf{Av}(s_i)$ and $\Delta(a_i, s_{i+1}) > 0$. Again, ρ_i refers to the i-th state visited by this particular run. A *path* of length k in $\mathcal{M}$ is a finite prefix $\varrho = s_1 a_1 \cdots a_{k-1} s_k$ of a run in G.

Strategies and plays. Intuitively, a strategy in an MDP $\mathcal{M}$ is a "recipe" to choose actions based on the observed events. Usually, a strategy is defined as a function $\sigma : (S\mathsf{A})^* S \to \mathcal{D}(\mathsf{A})$ that given a finite path ϱ, representing the history of a play, gives a probability distribution over the actions enabled in the last state. We adopt the slightly different, though equivalent [9, Sec. 6] definition from [15], which is more convenient for our setting.

Let M be a countable set of *memory elements*. A *strategy* is a triple $\sigma = (\sigma_u, \sigma_n, \alpha)$, where $\sigma_u : \mathsf{A} \times S \times \mathsf{M} \to \mathcal{D}(\mathsf{M})$ and $\sigma_n : S \times \mathsf{M} \to \mathcal{D}(\mathsf{A})$ are *memory update* and *next move* functions, respectively, and $\alpha \in \mathcal{D}(\mathsf{M})$ is the initial memory distribution. We require that, for all $(s, m) \in S \times \mathsf{M}$, the distribution $\sigma_n(s, m)$ assigns positive values only to actions available at s, i.e. $\operatorname{supp} \sigma_n(s, m) \subseteq \mathsf{Av}(s)$.

A *play* of $\mathcal{M}$ determined by a strategy σ is a Markov chain $\mathcal{M}^\sigma = (S^\sigma, \delta^\sigma, \mu_0^\sigma)$, where the set of states is $S^\sigma = S \times \mathsf{M} \times \mathsf{A}$, the initial distribution μ_0 is zero except for $\mu_0^\sigma(s_0, m, a) = \alpha(m) \cdot \sigma_n(s_0, m, a)$, and the transition probability from $s^\sigma = (s, m, a)$ to $s'^\sigma = (s', m', a')$ is $\delta^\sigma(s^\sigma, s'^\sigma) = \Delta(a, s') \cdot \sigma_u(a, s', m, m') \cdot \sigma_n(s', m', a')$. Hence, $\mathcal{M}^\sigma$ starts in a location chosen randomly according to α and σ_n. In state (s, m, a) the next action to be performed is a, hence the probability of entering s' is $\Delta(a, s')$. The probability of updating the memory to m' is $\sigma_u(a, s', m, m')$, and the probability of selecting a' as the next action is $\sigma_n(s', m', a')$. Since these choices are independent, and thus we obtain the product above.

Technically, $\mathcal{M}^\sigma$ induces a probability measure $\mathbb{P}^\sigma$ on S^σ. Since we mostly work with the corresponding runs in the original MDP, we overload $\mathbb{P}^\sigma$ to also refer to the probability measure obtained by projecting onto S. Further, "almost surely" etc. refers to happening with probability 1 according to $\mathbb{P}^\sigma$. The expected value of a random variable $X : \Omega \to \mathbb{R}$ is $\mathbb{E}^\sigma[X] = \int_\Omega X \, d\mathbb{P}^\sigma$.

A convex combinations of two strategies σ_1 and σ_2, written as $\sigma_\lambda = \lambda\sigma_1 + (1-\lambda)\sigma_2$, can be obtained by defining the memory as $\mathsf{M}_\lambda = \{1\} \times \mathsf{M}_1 \cup \{2\} \times \mathsf{M}_2$, randomly choosing one of the two strategies via the initial memory distribution α_λ and then following the chosen strategy. Clearly, we have that $\mathbb{P}^{\sigma_\lambda} = \lambda\mathbb{P}^{\sigma_1} + (1-\lambda)\mathbb{P}^{\sigma_2}$.

Strategy types. A strategy σ may use infinite memory M, and both σ_u and σ_n may randomize. The strategy σ is

- *deterministic-update*, if α is Dirac and the memory update function σ_u gives a Dirac distribution for every argument;
- *deterministic*, if it is deterministic-update and the next move function σ_n gives a Dirac distribution for every argument.

A *stochastic-update* strategy is a strategy that is not necessarily deterministic-update and *randomized* strategy is a strategy that is not necessarily deterministic. We also classify the strategies according to the size of memory they use. Important subclasses are *memoryless* strategies, in which M is a singleton, *n-memory* strategies, in which M has exactly n elements, and *finite-memory* strategies, in which M is finite.

End components. A tuple (T, B) where $\emptyset \neq T \subseteq S$ and $\emptyset \neq B \subseteq \bigcup_{t \in T} \mathsf{Av}(t)$ is an *end component* of the MDP $\mathcal{M}$ if (i) for all actions $a \in B$, $\Delta(a, s') > 0$ implies $s' \in T$; and (ii) for all states $s, t \in T$ there is a path $\varrho = s_1 a_1 \cdots a_{k-1} s_k \in (TB)^{k-1} T$ with $s_1 = s$, $s_k = t$. An end component (T, B) is a *maximal end component (MEC)* if T and B are maximal with respect to subset ordering. Given an MDP, the set of MECs is denoted by MEC. By abuse of notation, $s \in M$ refers to all states of a MEC M, while $a \in M$ refers to the actions.

Remark 1. *Computing the maximal end component (MEC) decomposition of an MDP, i.e. the computation of* MEC*, is in P [18].*

Remark 2. *For any MDP $\mathcal{M}$ and strategy σ, a run almost surely eventually stays in one MEC, i.e. $\mathbb{P}^\sigma[\bigcup_{M_i \in \mathsf{MEC}} \Diamond\Box M_i] = 1$ [31].*

2.2 Random variables on Runs

We introduce two standard random variables, assigning a value to each run of a Markov Chain or Markov Decision Process.

Weighted reachability. Let $T \subseteq S$ be a set of target states and $\mathbf{r} : T \mapsto \mathbb{Q}$ be a reward function. Define the random variable R^{r} as $R^{\mathsf{r}}(\rho) = \mathbf{r}(\min_i\{\rho_i \mid \rho_i \in T\})$, if such an i exists, and 0 otherwise. Informally, R^{r} assigns to each run the value of the first visited target state, or 0 if none. R^{r} is measurable and discrete, as S is finite [31]. Whenever we are dealing with weighted reachability, we assume w.l.o.g. that all target states are absorbing, i.e. for any $s \in T$ we have $\delta(s, s) = 1$ for MC and $\Delta(a, s) = 1$ for all $a \in \mathsf{Av}(s)$ for MDP.

Mean payoff (also known as *long-run average reward*, and *limit average reward*). Again, let $\mathbf{r} : S \mapsto \mathbb{Q}$ be a reward function. The mean payoff of a run ρ is the average reward obtained per step, i.e. $R^{\mathsf{m}}(\rho) = \liminf_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathbf{r}(\rho_i)$. The lim inf is necessary, since lim may not be defined in general. Further, R^{m} is measurable [31].

Remark 3. *There are several distinct definitions of "weighted reachability". The one chosen here primarily serves as foundation for the more general mean payoff.*

3 Introducing the Conditional Value-at-risk

In order to define our problem, we first introduce the general concept of *conditional value-at-risk* (CVaR), also known as *average value-at-risk*, *expected shortfall*, and *expected tail loss*. As already hinted, the CVaR of some real-valued random variable X and probability $p \in [0, 1]$ intuitively is the expectation below the worst p-quantile of X.

Let $X : \Omega \to \mathbb{R}$ be a random variable over the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The associated *cumulative density function* (CDF) $F_X : \mathbb{R} \to [0, 1]$ of X yields the probability of X being less than or equal to the given value r, i.e. $F_X(r) = \mathbb{P}(\{X(\omega) \leq r\})$. F is non-decreasing and right continuous with left limits (*càdlàg*).

The *value-at-risk* VaR_p is the worst p-quantile, i.e. a value v s.t. the probability of X attaining a value less than or equal to v is p:[3]

$$\mathrm{VaR}_p(X) := \sup\{r \in \mathbb{R} \mid F_X(r) \leq p\} \quad (\mathrm{VaR}_1(X) = \infty)$$

Then, with $v = \mathrm{VaR}_p(X)$, CVaR can be defined as [33]

$$\mathrm{CVaR}_p(X) := \mathbb{E}[X \mid X \leq v] = \frac{1}{p} \int_{(-\infty, v]} x \, dF_X,$$

with the corner cases $\mathrm{CVaR}_0 := \mathrm{VaR}_0$ and $\mathrm{CVaR}_1 = \mathbb{E}$.

Unfortunately, this definition only works as intended for continuous X, as shown by the following example.

[3] An often used, mostly equivalent definition is $\inf\{r \in \mathbb{R} \mid F_X(r) \geq p\}$. Unfortunately, this would lead to some complications later on. See [28, Sec. A.1] for details.

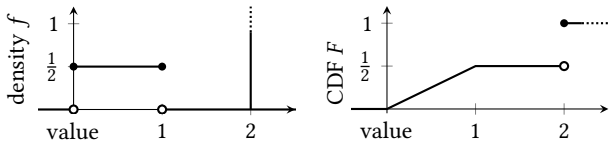

Figure 2. Distribution showing peculiarities of CVaR

Example 3.1. Consider a random variable X with a distribution as outlined in Fig. 2. For $p < \frac{1}{2}$, we certainly have $\mathrm{VaR}_p = 2p$. On the other hand, for any $p \in (\frac{1}{2}, 1)$, we get $\mathrm{VaR}_p = 2$. Consequently, the integral remains constant and CVaR_p would actually *decrease* for increasing p, not matching the intuition. △

General definition. As seen in Ex. 3.1, the previous definition breaks down when F_X is not continuous at the p-quantile and consequently $F_X(\mathrm{VaR}_p(X)) > p$. Thus, we handle the values at the threshold separately, similar to [34].

Definition 3.2. Let X be some random variable and $p \in [0, 1]$. With $v = \mathrm{VaR}_p(X)$, the CVaR of X is defined as

$$\mathrm{CVaR}_p(X) := \frac{1}{p}\left(\int_{(-\infty, v)} x \, dF_X + (p - \mathbb{P}[X < v]) \cdot v\right),$$

which can be rewritten as

$$\mathrm{CVaR}_p(X) = \tfrac{1}{p}\big(\mathbb{P}[X < v] \cdot \mathbb{E}[X \mid X < v] + (p - \mathbb{P}[X < v]) \cdot v\big).$$

The corner cases again are $\mathrm{CVaR}_0 := \mathrm{VaR}_0$, and $\mathrm{CVaR}_1 = \mathbb{E}$.

Since the degenerate cases of $p = 0$ and $p = 1$ reduce to already known problems, we exclude them in the following.

We demonstrate this definition on the previous example.

Example 3.3. Again, consider the random variable X from Ex. 3.1. For $\frac{1}{2} < p < 1$ we have that $\mathbb{P}[X < \mathrm{VaR}_p(X)] = \mathbb{P}[X < 2] = \frac{1}{2}$. The right hand side of the definition $(p - \mathbb{P}[X < \mathrm{VaR}_p(X)]) = p - \frac{1}{2}$ captures the remaining discrete probability mass which we have to handle separately. Together with $\int_{(-\infty, 2)} x \, dF_X = \frac{1}{4}$ we get $\mathrm{CVaR}_p(X) = \frac{1}{p}(\frac{1}{4} + (p - \frac{1}{2}) \cdot 2) = 2 - \frac{3}{4p}$. For example, with $p = \frac{3}{4}$, this yields the expected result $\mathrm{CVaR}_p(X) = 1$. △

Remark 4. *Recall that $\mathbb{P}[X < r]$ can be expressed as the left limit of F_X, namely $\mathbb{P}[X < r] = \lim_{r' \to^- r} F_X(r')$. Hence, $\mathrm{CVaR}_p(X)$ solely depends on the CDF of X and thus random variables with the same CDF also have the same CVaR.*

We say that F_1 *stochastically dominates* F_2 for two CDF F_1 and F_2, if $F_1(r) \leq F_2(r)$ for all r. Intuitively, this means that a sample drawn from F_2 is likely to be larger or equal to a sample from F_1. All three investigated operators ($\mathbb{E}$, CVaR, and VaR) are monotone w.r.t. stochastic dominance [28, Sec. A.1].

4 CVaR in MC and MDP: Problem statement

Now, we are ready to define our problem framework. First, we explain the types of building blocks for our queries, namely lower bounds on expectation, CVaR, and VaR. Formally, we consider the following types of constraints.

$$\mathrm{e} \leq \mathbb{E}(X) \qquad \mathrm{c} \leq \mathrm{CVaR}_{\mathrm{p}}(X) \qquad \mathrm{v} \leq \mathrm{VaR}_{\mathrm{q}}(X)$$

X is some real-valued random variable, assigning a payoff to each run. With these constraints, the classes of queries are denoted by

$$\mathrm{MDP}^{\mathbf{crit}}_{\mathbf{obj},\mathbf{dim}}$$

- **crit** $\subseteq \{\mathbb{E}, \mathrm{CVaR}, \mathrm{VaR}\}$ are the types of constraints,
- **obj** $\in \{\mathrm{r}, \mathrm{m}\}$ is the type of the objective function, either weighted reachability r or mean payoff m, and
- **dim** $\in \{\mathrm{single}, \mathrm{multi}\}$ is the dimensionality of the query.

We use d to denote the dimensions of the problem, $d = 1$ iff **dim** = single. As usual, we assume that all quantities of the input, e.g., probabilities of distributions, are rational.

An instance of these queries is specified by an MDP $\mathcal{M}$, a d-dimensional reward function $\mathbf{r} : S \to \mathbb{Q}^d$, and constraints from **crit**, given by vectors $\mathbf{e}, \mathbf{c}, \mathbf{v} \in (\mathbb{Q} \cup \{\bot\})^d$ and $\mathbf{p}, \mathbf{q} \in (0, 1)^d$. This implies that in each dimension there is at most one constraint per type. The presented methods can easily be extended to the more general setting of multiple constraints of a particular type in one dimension. The decision problem is to determine whether there exists a strategy σ such that *all* constraints are met.

Technically, this is defined as follows. Let X be the d-dimensional random variable induced by the objective **obj** and reward function $\mathbf{r}$, operating on the probability space of $\mathcal{M}^\sigma$. The strategy σ is a witness to the query iff for each dimension $j \in [d]$ we have that $\mathbb{E}[X_j] \geq \mathbf{e}_j$, $\mathrm{CVaR}_{\mathbf{p}_j}(X_j) \geq \mathbf{c}_j$, and $\mathrm{VaR}_{\mathbf{q}_j}(X_j) \geq \mathbf{v}_j$. Moreover, $\bot$ constraints are trivially satisfied.

For completeness sake, we also consider $\mathrm{MC}^{\mathbf{crit}}_{\mathbf{obj},\mathbf{dim}}$ queries, i.e. the corresponding problem on (finite state) Markov chains.

Notation. We introduce the following abbreviations. When dealing with an MDP $\mathcal{M}$, CVaR^σ_p denotes CVaR_p relative to the probability space over runs induced by the strategy σ. When additionally the random variable X (e.g., mean payoff) is clear from the context, we may write CVaR_p and CVaR^σ_p instead of $\mathrm{CVaR}_p(X)$ and $\mathrm{CVaR}^\sigma_p(X)$, respectively. We also define analogous abbreviations for VaR.

5 Single dimension

We show that all queries in one dimension are in P. Furthermore, our LP-based decision procedures directly yield a description of a witness strategy and allow for optimization objectives. We refer to the input constraints by e for expectation, (p, c) for CVaR, and (q, v) for VaR. Further, we use i for indices related to SCCs / MECs.

5.1 Weighted reachability

First, we show the simple result for Markov Chains, providing some insight in the techniques used in the MDP case.

Theorem 5.1. $\mathrm{MC}^{\{\mathbb{E}, \mathrm{CVaR}, \mathrm{VaR}\}}_{\mathrm{r},\mathrm{single}}$ *is in P.*

Proof. Let M be a *finite-state* Markov chain, $\mathbf{r}$ a reward function, and $T = \{b_1, \ldots, b_n\}$ the target set. Recall that all b_i are absorbing, hence single-state BSCCs. We obtain the stationary distribution p of M' in polynomial time by, e.g., solving a linear equation system [31]. With p, we can directly compute the CDF of R^{r} as $F_{R^{\mathrm{r}}}(v) = \sum_{b_i : \mathbf{r}(b_i) \leq v} p(b_i)$ and immediately decide the query. □

Let us consider the more complex case of MDP. We show a lower bound on the type of strategies necessary to realize **obj** = r queries with constraints on expectation and one of VaR or CVaR. We then continue to prove that this class of strategies is optimal. This characterization is used to derive a polynomial time decision procedure based on a linear program (LP) which immediately yields a witness strategy. Finally, when we deal with the mean payoff case in Sec. 5.2, we make use of the reasoning presented in this section.

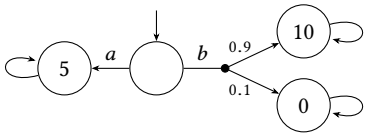

Figure 3. MDP used to show various difficulties of CVaR

Randomization is necessary for weighted reachability. In the following example, we present a simple MDP on which all deterministic strategies fail to satisfy specific constraints, while a straightforward randomizing one succeeds in doing so.

Example 5.2. Consider the MDP outlined in Fig. 3. The only nondeterminism is given by the choice in the initial state s_0. Hence, any strategy is characterised by the choice in that particular state. Let now σ_a and σ_b denote the deterministic strategies playing a and b in s_0, respectively. Clearly, σ_a achieves an expectation, $\mathrm{CVaR}_{0.05}^{\sigma_a}$, and $\mathrm{VaR}_{0.05}^{\sigma_a}$ of 5. On the other hand, σ_b obtains an expectation of 9 with $\mathrm{CVaR}_{0.05}^{\sigma_b}$ and $\mathrm{VaR}_{0.05}^{\sigma_b}$ equal to 0.

Thus, neither strategy satisfies the constraints $\mathsf{q} = \mathsf{p} = 0.05$, $\mathsf{e} = 6$, and $\mathsf{c} = 2$ (or $\mathsf{v} = 5$). This is the case even when the strategy has arbitrary (deterministic) memory at its disposal, since in the first step there is nothing to remember. Yet, $\sigma = \frac{3}{4}\sigma_a + \frac{1}{4}\sigma_b$ achieves $\mathbb{E} = \frac{3}{4}5 + \frac{1}{4}9 = 6 \geq \mathsf{e}$, $\mathrm{CVaR}_{\mathsf{p}} = 2.5 \geq \mathsf{c}$, and $\mathrm{VaR}_{\mathsf{q}} = 5 \geq \mathsf{v}$. △

Hence strategies satisfying an expectation constraint together with either a CVaR *or* VaR constraint may necessarily involve randomization in general. We prove that (i) under mild assumptions randomization actually is sufficient, i.e. no memory is required, and (ii) fixed memory may additionally be required in general.

Definition 5.3. Let $\mathcal{M}$ be an MDP with target set T and reward function $\mathbf{r}$. We say that $\mathcal{M}$ satisfies the *attraction assumption* if **A1**) the target set T is reached almost surely for any strategy, or **A2**) for all target state $s \in T$ we have $\mathbf{r}(s) \geq 0$.

Essentially, this definition implies that an optimal strategy never remains in a non-target MEC. This allows us to design memoryless strategies for the weighted reachability problem.

Theorem 5.4. *Memoryless randomizing strategies are sufficient for* $\mathrm{MDP}_{\mathrm{r,single}}^{\{\mathbb{E},\mathrm{VaR},\mathrm{CVaR}\}}$ *under the attraction assumption.*

Proof. Fix an MDP $\mathcal{M}$ and reward function $\mathbf{r}$. We prove that for any strategy σ there exists a memoryless, randomizing strategy σ' achieving at least the expectation, VaR, and CVaR of σ.

All target states $t_i \in T$ form single-state MECs, as we assumed that all target states are absorbing. Consequently, σ naturally induces a distribution over these s_i. Now, we apply [19, Theorem 3.2] to obtain a strategy σ' with $\mathbb{P}^{\sigma'}[\Diamond s_i] \geq \mathbb{P}^{\sigma}[\Diamond s_i]$ for all i.

With **A1**), we have $\sum p_i = 1$ and thus $\mathbb{P}^{\sigma'}[\Diamond t_i] = \mathbb{P}^{\sigma}[\Diamond t_i]$. Hence, σ' obtains the same CDF for the weighted reachability objective. Under **A2**), the CDF F' of strategy σ' stochastically dominates the CDF F of the original strategy σ, concluding the proof. □

Theorem 5.5. *Two-memory stochastic strategies (i.e. with both randomization and stochastic update) are sufficient for* $\mathrm{MDP}_{\mathrm{r,single}}^{\{\mathbb{E},\mathrm{VaR},\mathrm{CVaR}\}}$.

The proof is a simple application of the following Thm. 5.10, as weighted reachability is a special case of mean payoff. Together with an example for the lower bound it can be found in [28, Sec. A.2].

(1) All variables y_a, x_s, $\underline{x}_s$ are non-negative.
(2) Transient flow for $s \in S$:
$$\mathbb{1}_{s_0}(s) + \sum_{a \in \mathsf{A}} y_a \Delta(a,s) = \sum_{a \in \mathsf{Av}(s)} y_a + x_s$$
(3) Switching to recurrent behaviour:
$$\sum_{s \in T} x_s = 1$$
(4) VaR-consistent split:
$$\underline{x}_s = x_s \text{ for } s \in T_< \qquad \underline{x}_s \leq x_s \text{ for } s \in T_=$$
(5) Probability-consistent split:
$$\sum_{s \in T_\leq} \underline{x}_s = \mathsf{p}$$
(6) CVaR and expectation satisfaction:
$$\sum_{s \in T_\leq} \underline{x}_s \cdot \mathbf{r}(s) \geq \mathsf{p} \cdot \mathsf{c} \qquad \sum_{s \in T} x_s \cdot \mathbf{r}(s) \geq \mathsf{e}$$

Figure 4. LP used to decide weighted reachability queries given a guess t of $\mathrm{VaR}_{\mathsf{p}}$. $T_\sim := \{s \in T \mid \mathbf{r}(s) \sim t\}$, $\sim \in \{<, =, \leq\}$.

Inspired by [15, Fig. 3], we use the optimality result from Thm. 5.4 to derive a decision procedure for weighted reachability queries under the attraction assumptions based on the LP in Fig. 4.

To simplify the LP, we make further assumptions – see [28, Sec. A.2] for details. First, all MECs, including non-target ones, consist of a single state. Second, all MECs from which T is not reachable are considered part of T and have $\mathbf{r} = 0$ (similar to the "cleaned-up MDP" from [19]). Finally, we assume that the quantile-probabilities are equal, i.e. $\mathsf{p} = \mathsf{q}$. The LP can easily be extended to account for different values by duplicating the $\underline{x}_s$ variables and adding according constraints.

The central idea is to characterize randomizing strategies by the "flow" they achieve. To this end, Equality (2) essentially models Kirchhoff's law, i.e. inflow and outflow of a state have to be equal. In particular, y_a expresses the transient flow of the strategy as the expected total number of uses of action a. Similarly, x_s models the recurrent flow, which under our absorption assumption equals the probability of reaching s. Equality (3) ensures that all transient behaviour eventually changes into recurrent one.

In order to deal with our query constraints, Constraints (4) and (5) extract the worst p fraction of the recurrent flow, ensuring that the $\mathrm{VaR}_{\mathsf{p}}$ is at least t. Note that equality is not guaranteed by the LP; if $\underline{x}_s = x_s$ for all $s \in T_\leq$, we have $\mathrm{VaR}_{\mathsf{p}} > t$. Finally, Inequality (6) enforces satisfaction of the constraints.

Theorem 5.6. *Let $\mathcal{M}$ be an MDP with target states T and reward function $\mathbf{r}$, satisfying the attraction assumption. Fix the constraint probability $\mathsf{p} \in (0,1)$ and thresholds $\mathsf{e}, \mathsf{c} \in \mathbb{Q}$. Then, we have that*

1. *for any strategy σ satisfying the constraints, there is a $t \in \mathbf{r}(S)$ such that the LP in Fig. 4 is feasible, and*
2. *for any threshold $t \in \mathbf{r}(S)$, a solution of the LP in Fig. 4 induces a memoryless, randomizing strategy σ satisfying the constraints and $\mathrm{VaR}_{\mathsf{p}}^{\sigma} \geq t$.*

Proof. First, we prove for a strategy σ satisfying the constraints that there exists a $t \in \mathbf{r}(S)$ such that the LP is feasible. By Thm. 5.4, we may assume that σ is a memoryless randomizing strategy. From [19, Theorem 3.2], we get an assignment to the y_a's and x_s's satisfying Equalities (1), (2), and (3) such that $\mathbb{P}^{\sigma}[\Diamond s] = x_s$ for all target states

$s \in T$. Further, let $v = \mathrm{VaR}_{\mathsf{p}}^{\sigma}$ be the value-at-risk of the strategy. By definition of VaR, we have that $\mathbb{P}^{\sigma}[X < v] \leq p$.

Assume for now that $\mathbb{P}^{\sigma}[X < v] = p$, i.e. the probability of obtaining a value strictly smaller than v is exactly p. In this case, choose t to be the next smaller reward, i.e. $t = \max\{\mathbf{r}(s) < v\}$. We set $\underline{x}_s = x_s$ for all $s \in T_{\leq}$, satisfying Constraints (4) and (5).

Otherwise, we have $\mathbb{P}^{\sigma}[X < v] < p$. Now, some non-zero fraction of the probability mass at v contributes to the CVaR. Again, we set the values for $\underline{x}_s$ according to Constraint (4). The only degree of freedom are the values of $\underline{x}_s$ where $\mathbf{r}(s) = t$. There, we assign the values so that $\sum_{s \in T_{=}} \underline{x}_s = p - \sum_{s \in T_{<}} \underline{x}_s$, satisfying Equality (5).

It remains to check Inequality (6). For expectation, we have $\sum_{s \in T} x_s \cdot \mathbf{r}(s) = \sum_{s \in T} \mathbb{P}^{\sigma}[\Diamond s] \cdot \mathbf{r}(s) = \mathbb{E}^{\sigma}[R^{\mathsf{r}}] \geq \mathsf{e}$. For CVaR, notice that, due to the already proven Constraints (4) and (5), the side of Inequality (6) is equal to $\mathrm{CVaR}_{\mathsf{p}}^{\sigma}$ and thus at least c.

Second, we prove that a solution to the LP induces the desired strategy σ. Again by [19, Theorem 3.2], we get a memoryless randomizing strategy σ such that $\mathbb{P}^{\sigma}[\Diamond s] = x_s$ for all states $s \in T$. Then $\mathbb{E}^{\sigma}[R^{\mathsf{r}}] = \sum_{s \in T} \mathbb{P}^{\sigma}[\Diamond s] \cdot \mathbf{r}(s) = \sum_{s \in T} x_s \cdot \mathbf{r}(s) \geq \mathsf{e}$. Further,

$$\mathrm{CVaR}_p(R^{\mathsf{r}}) = \frac{1}{\mathsf{p}} \left(\sum\nolimits_{s:\mathbf{r}(s)<v} x_s \cdot \mathbf{r}(s) + (\mathsf{p} - \sum\nolimits_{s:\mathbf{r}(s)<v} x_s) \cdot v \right)$$

by definition. Now, we make a case distinction on $\underline{x}_s = x_s$ for all $s \in T_{=}$. If this is true, we have $v = \mathrm{VaR}_{\mathsf{p}}^{\sigma} = \min\{r \in \mathbf{r}(S) \mid r > t\}$, but $\mathbb{P}^{\sigma}[X < v] = p$. Consequently, $T_{\leq} = \{s \in T : \mathbf{r}(s) < v\}$ and $\sum_{s:\mathbf{r}(s)<v} x_s = p$. Otherwise, we have $v = t$ and consequently $T_{<} = \{s \mid \mathbf{r}(s) < v\}$. Inserting in the above equation immediately gives the result $\mathrm{CVaR}_p(R^{\mathsf{r}}) = \frac{1}{\mathsf{p}} \sum_{s \in T_{\leq}} \mathbf{r}(s) \cdot \underline{x}_s$. □

The linear program requires to know the VaR_p^{σ} beforehand, which in turn clearly depends on the chosen strategy. Yet, there are only linearly many values the random variable R^{r} attains. Thus we can simply try to find a solution for all potential values of VaR_p^{σ}, i.e. $\{r \in \mathbf{r}(S)\}$, yielding a polynomial time solution.

Corollary 5.7. $\mathsf{MDP}^{\{\mathbb{E},\mathrm{VaR},\mathrm{CVaR}\}}_{\mathsf{r,single}}$ *is in P.*

Proof. Under the attraction assumption, this follows directly from Thm. 5.6. In general, the reduction to mean payoff used by Thm. 5.5 and the respective result from Cor. 5.11 show the result. □

5.2 Mean payoff

In this section, we investigate the case of **obj** = m. Again, the construction for MC is considerably simple, yet instructive for the following MDP case.

Theorem 5.8. $\mathsf{MC}^{\{\mathbb{E},\mathrm{VaR},\mathrm{CVaR}\}}_{\mathsf{m,single}}$ *is in P.*

Proof sketch. For each BSCC B_i, we obtain its expected mean payoff $r_i = \mathbb{E}[R^{\mathsf{m}} \mid B_i]$ through, e.g., a linear equation system [31]. Almost all runs in B_i achieve this mean payoff and thus the corresponding random variable is discrete. We reduce the problem to weighted reachability by using the known reformulation

$$\mathbb{P}[R^{\mathsf{m}} = c] = \sum\nolimits_{B_i : r_i = c} P[\Diamond B_i].$$

We replace each of these BSCCs by a representative b_i to obtain M'. Define the set of target states $T = \{b_i\}$ and the reachability reward function $\mathbf{r}'(b_i) = r_i$. By applying the approach of Thm. 5.1, we obtain the expectation, VaR, and CVaR for reachability in M' which by construction coincides with the respective values for mean payoff in M. □

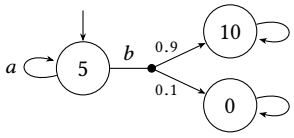

Figure 5. Memory is necessary for mean payoff queries

For the MDP case, recall that simple expectation maximization of mean payoff can be reduced to weighted reachability [2] and deterministic, memoryless strategies are optimal [31]. Yet, solving a conjunctive query involving either VaR or CVaR needs more powerful strategies than in the weighted reachability case of Thm. 5.4. Nevertheless, we show how to decide these queries in P.

Randomization and memory is necessary for mean payoff. A simple modification of the MDP in Fig. 3 yields an MDP where both randomization and memory is required to satisfy the constraints of the following example.

Example 5.9. Consider the MDP presented in Fig. 5. There, the same constraints as before, i.e. q = p = 0.05, e = 6, and c = 2 (or v = 5), can only be satisfied by strategies with both memory and randomization. Clearly, a pure strategy can only satisfy either of the two constraints again. But now a memoryless randomizing strategy also is insufficient, too, since any non-zero probability on action b leads to almost all runs ending up on the right side of the MDP, hence yielding a $\mathrm{CVaR}_{\mathsf{p}}$ and $\mathrm{VaR}_{\mathsf{q}}$ of 0. Instead, a stochastic strategy with $\mathsf{M} = \{a, b\}$ can simply choose $\alpha = \{a \mapsto \frac{3}{4}, b \mapsto \frac{1}{4}\}$ and play the corresponding action indefinitely, satisfying the constraints. △

We prove that this bound actually is tight, i.e. that, given stochastic memory update, two memory elements are sufficient.

Theorem 5.10. *Two-memory stochastic strategies (i.e. with both randomization and stochastic update) are sufficient for* $\mathsf{MDP}^{\{\mathbb{E},\mathrm{VaR},\mathrm{CVaR}\}}_{\mathsf{m,single}}$.

Proof. Let σ be a strategy on an MDP $\mathcal{M}$ with reward function $\mathbf{r}$. We construct a two-memory stochastic strategy σ' achieving at least the expectation, VaR, and CVaR of σ.

First, we obtain a memoryless deterministic strategy σ_{opt} which obtains the maximal possible mean payoff in each MEC [31]. We then apply the construction of [9, Proposition 4.2] (see also [15, Lemma 5.7]), where the ξ is our σ_{opt}. (Technically, this can be ensured by choosing the constraints of the LP L according to σ_{opt}.)

Intuitively, this constructs a two-memory strategy σ' on $\mathcal{M}$ behaving as follows. Initially, σ' remains in each MEC with the same probability as σ, i.e. $\mathbb{P}^{\sigma'}[\Diamond\Box M_i] = \mathbb{P}^{\sigma}[\Diamond\Box M_i]$ by following a memoryless "searching" strategy and stochastically switching its memory state to "remain". Once in the "remain" state, the behaviour of the optimal strategy σ_{opt} is implemented.

Clearly, (i) both strategies remain in a particular MEC with the same probability, and (ii) σ' obtains as least as much value in each MEC as σ. Hence the CDF induced by σ' stochastically dominates the one of σ, concluding the proof. □

This immediately gives us a polynomial time decision procedure.

Corollary 5.11. $\mathsf{MDP}^{\{\mathbb{E},\mathrm{VaR},\mathrm{CVaR}\}}_{\mathsf{m,single}}$ *is in P.*

Furthermore, we can use results of [15, Lemma 16] to trade the stochastic update for more memory.

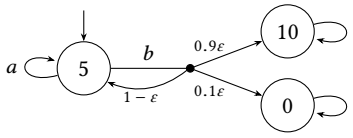

Figure 6. Exponential memory is necessary for mean payoff when only deterministic update is allowed.

Corollary 5.12. *Stochastic strategies with finite, deterministic memory are sufficient for* $\mathsf{MDP}^{\{\mathbb{E},\mathrm{VaR},\mathrm{CVaR}\}}_{\mathsf{m},\mathsf{single}}$.

Deterministic strategies may require exponential memory. As sources of randomness are not always available, one might ask what can be hoped for when only determinism is allowed. As already shown in Ex. 5.2, randomization is required in general. But even if some deterministic strategy is sufficient, it may require memory exponential in the size of the input, even in an MDP with only 3 states. We show this in the following example.

Example 5.13. Consider the MDP outlined in Fig. 6 together with the constraints q = p = 0.05, e = 6, and c = 2 (or v = 5). Again, any optimal strategy needs a significant part of runs to go to the right side in order to satisfy the expectation constraint. Yet, any strategy can only "move" a small fraction of the runs there in each step. In particular, after k steps, the right side is only reached with probability at most $1-(1-\varepsilon)^k$. When choosing $\varepsilon = 2^{-n}$, which needs $\Theta(n)$ bits to encode, a deterministic strategy requires $k \geq c/\log(1-2^{-n}) \in O(2^n)$ memory elements to count the number of steps. The same holds true for any deterministic-update strategy.

On the other hand, a strategy with stochastic memory update can encode this counting by switching its state with a small probability after each step. For example, a strategy switching with probability $p = 3\varepsilon$ from "play b" to "play a" satisfies the constraint. △

5.3 Single constraint queries

In this section, we discuss an important sub-case of the single-dimensional case, namely queries with only a single constraint, i.e. $|\mathbf{crit}| = 1$. We show that deterministic memoryless strategies are sufficient in this case.

One might be tempted to use standard arguments and directly conclude this from the results of Thm. 5.4 as follows. Recall that this theorem shows that memoryless, randomizing strategies are sufficient; and that any such strategy can be written as finite convex combination of memoryless, deterministic strategies. Most constraints, for example expectation or reachability, behave linearly under convex combination of strategies, e.g., $\mathbb{E}^{\sigma_\lambda}(X) = \lambda\mathbb{E}^{\sigma_1}[X] + (1-\lambda)\mathbb{E}^{\sigma_2}[X]$. Consequently, for an optimal memoryless strategy, there is a deterministic witness, which in turn also is optimal.

Surprisingly, this assumption is not true for CVaR. On the contrary, the CVaR of a convex combination of strategies might be strictly worse than the CVaRs of either strategy, as shown in the following example. We prove a slightly weaker property of CVaR which eventually allows us to apply similar reasoning.

Example 5.14. Recall the MDP in Fig. 3 and let $p = 0.05$. As previously shown, $\mathrm{CVaR}_p^{\sigma_a} = 5$ and $\mathrm{CVaR}_p^{\sigma_b} = 0$, but the mixed strategy $\sigma_\lambda = \frac{1}{2}\sigma_a + \frac{1}{2}\sigma_b$ achieves $\mathrm{CVaR}_p^{\sigma_\lambda} = 0$ instead of the convex combination $\frac{1}{2}5 + \frac{1}{2}0 = 2.5$.

For $p = 0.2$, we have $\mathrm{CVaR}_p^{\sigma_a} = \mathrm{CVaR}_p^{\sigma_b} = 5$. Yet, *any* non-trivial convex combination of the two strategies yields a CVaR_p less than 5. See [28, Sec. A.1] for more details. With according constraints, this effectively can force an optimal strategy to choose between a or b. This observation is further exploited in the NP-hardness proof of the multi-dimensional case in Sec. 6. △

Since CVaR considers the worst events, the CVaR of a combination intuitively cannot be better than the combination of the respective CVaRs. We prove this intuition in the general setting, where instead of a convex combination of strategies we consider a mixture of two random variables.

Lemma 5.15. $\mathrm{CVaR}_p(X)$ *is convex in* X *for fixed* $p \in (0,1)$*, i.e. for random variables* X_1, X_2 *and* $\lambda \in [0,1]$

$$\mathrm{CVaR}_p(\lambda X_1 + (1-\lambda)X_2) \leq \lambda\,\mathrm{CVaR}_p(X_1) + (1-\lambda)\,\mathrm{CVaR}_p(X_2).$$

The proof can be found in [28, Sec. A.1]. This result allows us to apply the ideas outlined in the beginning of the section.

Theorem 5.16. *For any* $\mathbf{obj} \in \{\mathsf{r},\mathsf{m}\}$*, deterministic memoryless strategies are sufficient for* $\mathsf{MDP}^{\mathbf{crit}}_{\mathbf{obj},\mathsf{single}}$ *when* $|\mathbf{crit}| = 1$.

Proof. This is known for $\mathbf{crit} = \{\mathbb{E}\}$ [31] and $\mathbf{crit} = \{\mathrm{VaR}\}$ [21].

For CVaR, observe that the convex combination of deterministic strategies cannot achieve a better CVaR than the best strategy involved in the combination (see Lem. 5.15). This immediately yields the result for $\mathbf{obj} = \mathsf{r}$ through Thm. 5.4. For $\mathbf{obj} = \mathsf{m}$, we exploit the approach of Thm. 5.10. Recall that there we obtained a two-memory strategy σ'. Both randomization and stochastic update are used solely to distribute the runs over all MECs accordingly. By the above reasoning, for each MEC it is sufficient to either almost surely remain there or leave it. This behaviour can be implemented by a deterministic memoryless strategy on the original MDP. □

6 Multiple Dimensions

In this section, we deal with multi-dimensional queries. We continue to use i for indices related to MECs and further use j for dimension indices. First, we show that the Markov Chain case does not significantly change.

Theorem 6.1. *For any* $\mathbf{obj} \in \{\mathsf{r},\mathsf{m}\}$, $\mathsf{MC}^{\{\mathbb{E},\mathrm{VaR},\mathrm{CVaR}\}}_{\mathbf{obj},\mathsf{multi}}$ *is in P.*

Proof. Similarly to the single-dimensional case, we decide each constraint in each dimension separately, using our previous results. The query is satisfied iff each of the constraints is satisfied. □

6.1 NP-Hardness of reachability and mean payoff

For the MDP on the other hand, multiple dimensions add significant complexity. In the following, we show that already the weighted reachability problem with multiple dimensions and only CVaR constraints, i.e. $\mathsf{MDP}^{\{\mathrm{CVaR}\}}_{\mathsf{r},\mathsf{multi}}$, is NP-hard. This result directly transfers to mean payoff, i.e. $\mathbf{obj} = \mathsf{m}$. Recall that in contrast $\mathsf{MDP}^{\{\mathbb{E}\}}_{\mathsf{r},\mathsf{multi}}$ and even $\mathsf{MDP}^{\{\mathbb{E},\mathrm{VaR}_0\}}_{\mathsf{r},\mathsf{multi}}$, i.e. constraints on the expectation and ensuring that almost all runs achieve a given threshold, are in P [15].

Theorem 6.2. *For any* $\mathbf{obj} \in \{\mathsf{r},\mathsf{m}\}$, $\mathsf{MDP}^{\{\mathrm{CVaR}\}}_{\mathbf{obj},\mathsf{multi}}$ *is NP-hard (when the dimension* d *is a part of the input).*

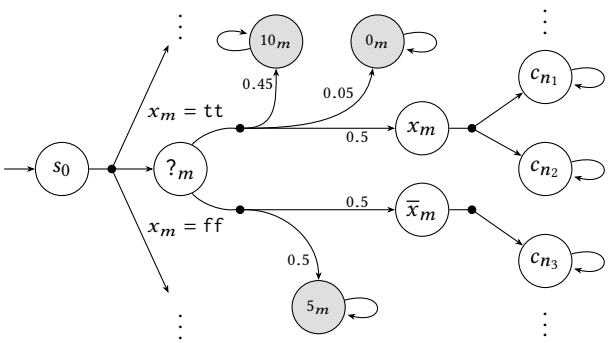

Figure 7. Gadget for variable x_m. Uniform transition probabilities are omitted for readability.

Proof. We prove hardness by reduction from 3-SAT. The core idea is to utilize observations from Fig. 3 and Ex. 5.14, namely that CVaR constraints can be used to enforce a deterministic choice.

Let $\{C_n\}$ be a set of N clauses with M variables x_m and set the dimensions $d = N+M$. By abuse of notation, n refers to the dimension of clause C_n and m to the one of variable x_m, respectively.

The gadget for the reduction is outlined in Fig. 7. Observe that, due to the structure of the MDP, we have that $R^{\mathsf{r}} = R^{\mathsf{m}}$.

Overall, the reduction works as follows. Initially, a state $?_m$, representing the variable x_m, is chosen uniformly. In this state, the strategy is asked to give the valuation of x_m through the actions "x_m = tt" or "x_m = ff". As seen in Ex. 5.14, the structure of the shaded states can be used to enforces a deterministic choice between the two actions. Particularly, in dimension m we require $\mathrm{CVaR}_p \geq 5$ for $p = \frac{M-1}{M} + \frac{1}{M} \cdot 0.5 \cdot 0.2$. Since all other gadgets yield 0 in dimension m and only half of the runs going through $?_m$ end up in the shaded area, this corresponds to Ex. 5.14, where $p = 0.2$.

Once in either state x_m or $\overline{x}_m$, a state c_n corresponding to a clause C_n satisfied by this assignment is chosen uniformly. In the example gadget, we would have $x_m \in C_{n_1} \cap C_{n_2}$, and $\overline{x}_m \in C_{n_3}$. We set the reward of c_n to 1_n. Then a clause c_n is satisfied under the assignment if the state c_n is visited with positive probability, e.g. if $\mathrm{CVaR}_1 \geq \frac{1}{M} \cdot 0.5 \cdot \frac{1}{N}$. Clearly, a satisfying assignment exists iff a strategy satisfying these constraints exists. □

6.2 NP-completeness and strategies for reachability

For weighted reachability, we prove that the previously presented bound is tight, i.e. that the weighted reachability problem with multiple dimensions and CVaR constraints is NP-complete when d is part of the input and P otherwise. First, we show that the strategy bounds of the single dimensional case directly transfer. Intuitively, this is the case since only the steady state distribution over the target set T is relevant, independent of the dimensionality.

Theorem 6.3. *Two-memory stochastic strategies (i.e. with both randomization and stochastic update) are sufficient for* $\mathsf{MDP}^{\{\mathbb{E},\mathrm{VaR},\mathrm{CVaR}\}}_{\mathsf{r},\mathsf{multi}}$. *Moreover, if* $\mathbf{r}_j(s) \geq 0$ *for all* $s \in T$ *and* $j \in [d]$, *then memoryless randomizing strategies are sufficient.*

Proof. Follows directly from the reasoning used in the proofs of Thm. 5.10 and Thm. 5.4. □

(1) All variables y_a, x_s, $\underline{x}^j_s$ are non-negative.
(4) VaR-consistent split for $j \in [d]$:

$$\underline{x}^j_s = x_s \text{ for } s \in T^j_{<} \qquad \underline{x}^j_s \leq x_s \text{ for } s \in T^j_{=}$$

(5) Probability-consistent split for $j \in [d]$:

$$\sum\nolimits_{s \in T^j_{\leq}} \underline{x}^j_s = \mathbf{p}_j$$

(6) CVaR and expectation satisfaction for $j \in [d]$:

$$\sum\nolimits_{s \in T^j_{\leq}} \underline{x}_s \cdot \mathbf{r}(s) \geq \mathrm{p}_j \cdot \mathbf{c}_j \qquad \sum\nolimits_{s \in T} x_s \cdot \mathbf{r}_j(s) \geq \mathbf{e}_j$$

Figure 8. LP used to decide multi-dimensional weighted reachability queries given a guess $\mathbf{t}$ of $\mathrm{VaR}_{\mathrm{p}_j}$. Equalities (2) and (3) are as in Fig. 4, $T^j_{\sim} := \{s \in T \mid \mathbf{r}_j(s) \sim \mathbf{t}_j\}$, $\sim \in \{<, =, \leq\}$.

Theorem 6.4. $\mathsf{MDP}^{\{\mathbb{E},\mathrm{VaR},\mathrm{CVaR}\}}_{\mathsf{r},\mathsf{multi}}$ *is in NP if* d *is a part of the input; moreover, it is in P for any fixed* d.

Proof sketch. To prove containment, we guess the VaR threshold vector $\mathbf{t}$ out of the set of potential ones, namely $\{r \mid \exists i \in [d], s \in T . \mathbf{r}_i(s) = r\}^d$ and use an LP to verify the solution. We again assume that each MEC can reach the target set and is single-state, as we did for Fig. 4. The arguments used to resolve this assumption are still applicable in the multi-dimensional setting. The LP consists of the flow Equalities (2) and (3) from the LP in Fig. 4 together with the modified (In)Equalities (4)-(6) as shown in Fig. 8.

The difference is that we extract the worst fraction of the flow *in each dimension*. Consequently, we have d instances of each $\underline{x}_s$ variable, namely $\underline{x}^j_s$. The number of possible guesses $\mathbf{t}$ is bounded by $|T|^d$ and thus the guess is of polynomial length. For a fixed d the bound itself is polynomial and hence, as previously, we can try out all vectors. □

6.3 Upper bounds of mean payoff

In this section, we provide an upper bound on the complexity of mean-payoff queries. Strategies in this context are known to have higher complexity.

Proposition 6.5 ([9]). *Infinite memory is necessary for* $\mathsf{MDP}^{\{\mathbb{E}\}}_{\mathsf{m},\mathsf{multi}}$.

Note that this directly transfers to $\mathsf{MDP}^{\{\mathrm{CVaR}\}}_{\mathsf{m},\mathsf{multi}}$, as $\mathrm{CVaR}_1 = \mathbb{E}$. However, closing gaps between lower and upper bounds for the mean payoff objective is notoriously more difficult. For instance, $\mathsf{MDP}^{\{\mathrm{VaR}\}}_{\mathsf{m},\mathsf{multi}}$ is known to be in EXP, but not even known to be NP-hard, neither is $\mathsf{MDP}^{\{\mathbb{E},\mathrm{VaR}\}}_{\mathsf{m},\mathsf{multi}}$. Since we have proven that $\mathsf{MDP}^{\{\mathrm{CVaR}\}}_{\mathsf{m},\mathsf{multi}}$ is NP-hard, we can expect that obtaining the matching NP upper bound will be yet more difficult. The fundamental difference of the multi-dimensional mean-payoff case is that the solutions within MECs cannot be pre-computed, rather non-trivial trade-offs must be considered. Moreover, the trade-offs are not "local" and must be synchronized over all the MECs, see [15] for details.

We now observe that, as opposed to quantile queries, i.e. VaR constraints, the behaviour inside each MEC can be assumed to be quite simple. Our results primarily rely on [16] and use a similar notation. In particular, given a run ρ, $\mathrm{Freq}_a(\rho)$ yields the average frequency of action a, i.e. $\mathrm{Freq}_a(\rho) := \liminf_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{1}_a(a_t)$, where a_t refers to the action taken by ρ in step t.

Definition 6.6. A strategy σ is *MEC-constant* if for all $M_i \in \mathsf{MEC}$ with $\mathbb{P}^\sigma[\Diamond\Box M_i] > 0$ and all $j \in [d]$ there is a $v \in \mathbb{R}$ such that $\mathbb{P}^\sigma[R^{\mathrm{m}}_j = v \mid \Diamond\Box M_i] = 1$.

Lemma 6.7. *MEC-constant strategies are sufficient for* $\mathsf{MDP}^{\{\mathbb{E},\mathrm{CVaR}\}}_{\mathrm{m,multi}}$.

Proof. Fix an MDP $\mathcal{M}$ with MECs $\mathsf{MEC} = \{M_1, \dots, M_n\}$, reward function $\mathbf{r}$ and a strategy σ. Further, define $p_i = \mathbb{P}^\sigma[\Diamond\Box M_i]$. We construct a strategy σ' so that (i) $\mathbb{P}^{\sigma'}[\Diamond\Box M_i] = p_i$ for all M_i, and (ii) all behaviours of σ on a MEC M_i are "mixed" into each run on M_i, making it MEC-constant.

We first define the mixing strategies σ_i, achieving point (ii). By [16, Sec. 4.1], there are frequencies $(x_a)_{a\in\mathsf{A}}$ which

- satisfy $\sum_{a\in A} x_a \cdot \Delta(a,s) = \sum_{a\in\mathsf{Av}(s)} x_a$ for all $s \in S$,
- for each action a we have $\mathbb{E}^\sigma[\mathrm{Freq}_a] \le x_a$, and
- $\sum_{a\in\mathsf{A}\cap M_i} x_a = p_i$.

By [16, Cor. 5.5], there is a (Markov) strategy σ_i on M_i where

$$\mathbb{P}^{\sigma_i}\left[\mathrm{Freq}_a = x_a/p_i\right] = 1.$$

Consequently, σ_i is almost surely constant on M_i w.r.t. R^{m}. We apply the reasoning used in the proof of Thm. 5.10 to obtain the combined strategy σ' which achieves point (i) and switches to σ_i upon remaining in M_i.

Now, fix any $j \in [d]$, $M_i \in \mathsf{MEC}$, and $p, q \in (0,1)$. We have that $\mathbb{E}^{\sigma_i}[\mathrm{Freq}_a \mid \Diamond\Box M_i] \ge \mathbb{E}^\sigma[\mathrm{Freq}_a \mid \Diamond\Box M_i]$ by construction. Consequently, $\mathbb{E}^{\sigma'}(R^{\mathrm{m}}_j) \ge \mathbb{E}^\sigma(R^{\mathrm{m}}_j)$.

Since σ' is MEC-constant, we have $\mathrm{CVaR}^{\sigma'}_p(R^{\mathrm{m}}_j \mid \Diamond\Box M_i) = \mathbb{E}^{\sigma'}[R^{\mathrm{m}}_j \mid \Diamond\Box M_i]$. Further, by $\mathbb{E}^\sigma[\mathrm{Freq}_a \mid \Diamond\Box M_i] \cdot p_i \le \mathbb{E}^{\sigma_i}[\mathrm{Freq}_a]$ for all a, we get $\mathbb{E}^\sigma[R^{\mathrm{m}}_j \mid \Diamond\Box M_i] \le \mathbb{E}^{\sigma_i}[R^{\mathrm{m}}_j]$. So, $\mathrm{CVaR}^{\sigma_i}_p(R^{\mathrm{m}}_j) = \mathbb{E}^{\sigma_i}[R^{\mathrm{m}}_j] \ge \mathbb{E}^\sigma[R^{\mathrm{m}}_j \mid \Diamond\Box M_i] \ge \mathrm{CVaR}^\sigma_q(R^{\mathrm{m}}_j \mid \Diamond\Box M_i)$, as $\mathrm{CVaR} \le \mathbb{E}$.

Finally, we apply this inequality together with property (i), obtaining $\mathrm{CVaR}^\sigma_p(R^{\mathrm{m}}_j) \le \mathrm{CVaR}^{\sigma'}_p(R^{\mathrm{m}}_j)$ by [28, Thm. A.4] □

We utilize this structural property to design a linear program for these constraints. However, similarly to the previously considered LPs, it relies on knowing the VaR for each $\mathrm{CVaR_p}$ constraint. Due to the non-linear behaviour of CVaR, the classical techniques do not allow us to conclude that VaR is polynomially sized and thus we do not present the "matching" NP upper bound, but a PSPACE upper bound, which we achieve as follows.

Theorem 6.8. $\mathsf{MDP}^{\{\mathbb{E},\mathrm{CVaR}\}}_{\mathrm{m,multi}}$ *is in PSPACE.*

Proof sketch. We use the existential theory of the reals, which is NP-hard and in PSPACE [12], to encode our problem. The VaR vector $\mathbf{t}$ is existentially quantified and the formula is a polynomially sized program with constraints linear in VaR's and linear in the remaining variables. This shows the complexity result.

The details of the procedure are as follows. For each $j \in [d]$, we use the existential theory of reals to guess the achieved VaR $\mathbf{t} = \mathrm{VaR}_{\mathbf{p}_j}$. Further, we non-deterministically obtain the following polynomially-sized information (or deterministically try out all options in PSPACE). For each $j \in [d]$ and for each MEC M_i, we guess if the value achieved in M_i is at most (denoted $M_i \in \mathsf{MEC}^j_{\le}$) or above (denoted $M_i \in \mathsf{MEC}^j_{>}$) the respective $\mathbf{t}_j$, and exactly one MEC $M^j_=$, which achieves a value equal to it. Given these guesses, we check whether the LP in Fig. 9 has a solution.

(1) All variables y_a, y_s, x_a, x_s are non-negative.
(2) Transient flow for $s \in S$:
$$\mathbb{1}_{s_0}(s) + \sum_{a\in\mathsf{A}} y_a \cdot \Delta(a,s) = \sum_{a\in\mathsf{Av}(s)} y_a + y_s$$
(3) Probability of switching in a MEC is the frequency of using its actions for $M_i \in \mathsf{MEC}$:
$$\sum_{s\in M_i} y_s = \sum_{a\in M_i} x_a$$
(4) Recurrent flow for $s \in S$:
$$x_s = \sum_{a\in\mathsf{A}} x_a \cdot \Delta(a,s) = \sum_{a\in\mathsf{Av}(s)} x_a$$
(5) CVaR and expectation satisfaction for $j \in [d]$:
$$\sum_{s\in S^j_{\le}} x_s \cdot \mathbf{r}_j(s) + \left(\mathbf{p}_j - \sum_{s\in S^j_{\le}} x_s\right) \cdot \mathbf{t}_j \ge \mathbf{p}_j \cdot \mathbf{c}_j$$
$$\sum_{s\in S} x_s \cdot \mathbf{r}_j(s) \ge \mathbf{e}_j$$
(6) Verify MEC classification guess for $j \in [d]$:
$$\sum_{s\in M^j_{\le}} x_s \cdot \mathbf{r}_j(s) \le \mathbf{t}_j \quad \text{for } M^j_{\le} \in \mathsf{MEC}^j_{\le} \cup \{M^j_=\}$$
$$\sum_{s\in M^j_{\ge}} x_s \cdot \mathbf{r}_j(s) \ge \mathbf{t}_j \quad \text{for } M^j_{\ge} \in \mathsf{MEC}^j_{>} \cup \{M^j_=\}$$
(7) Verify VaR guess for $j \in [d]$:
$$\sum_{s\in S^j_{\le}} x_s \le \mathbf{p}_j \qquad \sum_{s\in S^j_{\le}\cup M^j_=} x_s \ge \mathbf{p}_j$$

Figure 9. LP used to decide multi-dimensional mean-payoff queries given a guess $\mathbf{t}$ of $\mathrm{VaR}_{\mathbf{p}_j}$ and MEC classification $\mathsf{MEC}^j_{\le}$, $M^j_=$, and $\mathsf{MEC}^j_{>}$. $S^j_\sim := \{s \in S \mid s \in M \text{ and } M \in \mathsf{MEC}^j_\sim\}$, $\sim \in \{\le, >\}$.

Equations (1)-(4) describe the transient flow like the previous LP's and, additionally, the recurrent flow like in [31, Sec. 9.3] or [9, 16, 19]. This addition is needed, since now our MECs are not trivial, i.e. single state. Again, Inequalities (5) verify that the CVaR and expectation constraints are satisfied. Finally, Inequalities (6) and (7) verify the previously guessed information, i.e. the VaR vector and the MEC classification.

Using the very same techniques, it is easy to prove that solutions to the LP correspond to satisfying strategies and vice versa. In particular, Inequalities (6) and (7) directly make use of the MEC-constant property of Lem. 6.7. □

While MEC-constant strategies are sufficient for $\mathbb{E}$ with CVaR, in contrast, they are not even for just $\mathsf{MDP}^{\{\mathrm{VaR}\}}_{\mathrm{m,multi}}$ [15, Ex.22]. Consequently, only an exponentially large LP is known for $\mathsf{MDP}^{\{\mathrm{VaR}\}}_{\mathrm{m,multi}}$. We can combine all the objective functions together as follows:

Theorem 6.9. $\mathsf{MDP}^{\{\mathbb{E},\mathrm{VaR},\mathrm{CVaR}\}}_{\mathrm{m,multi}}$ *is in EXPSPACE.*

Proof sketch. We proceed exactly as in the previous case, but now the flows in Equality (4) are split into exponentially many flows, depending on the set of dimensions where they achieve the given VaR threshold, see LP L in [15, Fig. 4]. The resulting size of the program is polynomial in the size of the system and exponential in d. Hence the call to the decision procedure of the existential theory of reals results in the EXPSPACE upper bound. □

Table 1. Schematic summary of known and new results. Strategies are abbreviated by "C/n-M", where C is either *D*eterministic or *R*andomizing, n is the size of the memory, and M is either *D*etereministic or *S*tochastic *MEM*ory.

dim	single		multi				
obj	any		r	m			
crit	$\lvert\mathbf{crit}\rvert = 1$	$\lvert\mathbf{crit}\rvert \geq 2$	CVaR $\in$ **crit**	$\{\mathbb{E}, \mathrm{VaR}_0\}$	{VaR}	{CVaR}, {CVaR, $\mathbb{E}$}	{$\mathbb{E}$, CVaR, VaR}
Complex.	P		NP-c., P for fixed d	P	EXP	NP-h., PSPACE	NP-h., EXPSPACE
Strat.	D/1-MEM	R/2-SMEM	R/2-SMEM			R/∞-DMEM	

7 Conclusion

We introduced the conditional value-at-risk for Markov decision processes in the setting of classical verification objectives of reachability and mean payoff. We observed that in the single dimensional case the additional CVaR constraints do not increase the computational complexity of the problems. As such they provide a useful means for designing risk-averse strategies, at no additional cost. In the multidimensional case, the problems become NP-hard. Nevertheless, this may not necessarily hinder the practical usability. Our results are summarized in Table 1.

We conjecture that the VaR's for given CVaR constraints are polynomially large numbers. In that case, the provided algorithms would yield NP-completeness for $\mathsf{MDP}^{\{\mathrm{CVaR}\}}_{\mathsf{m,multi}}$ and EXPTIME-containment for $\mathsf{MDP}^{\{\mathbb{E},\mathrm{VaR},\mathrm{CVaR}\}}_{\mathsf{m,multi}}$, where the exponential dependency is only on the dimension, not the size of the system.

Acknowledgments

This research has been partially supported by the Czech Science Foundation grant No. 18-11193S and the German Research Foundation (DFG) project KR 4890/2 "Statistical Unbounded Verification" (383882557). We thank Vojtěch Forejt for bringing up the topic of CVaR and the initial discussions with Jan Krčál and wish them both happy life in industry. We also thank Michael Luttenberger and the anonymous reviewers for insightful comments and valuable suggestions.

References

[1] Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. 1999. Coherent Measures of Risk. *Mathematical Finance* 9, 3 (1999), 203–228.
[2] Pranav Ashok, Krishnendu Chatterjee, Przemyslaw Daca, Jan Křetínský, and Tobias Meggendorfer. 2017. Value Iteration for Long-Run Average Reward in Markov Decision Processes. In *CAV (LNCS)*, Vol. 10426. Springer, 201–221.
[3] Christel Baier, Marcus Daum, Clemens Dubslaff, Joachim Klein, and Sascha Klüppelholz. 2014. Energy-Utility Quantiles. In *NFM (LNCS)*, Vol. 8430. Springer, 285–299.
[4] Christel Baier, Clemens Dubslaff, and Sascha Klüppelholz. 2014. Trade-off analysis meets probabilistic model checking. In *CSL-LICS*. ACM, 1:1–1:10.
[5] Christel Baier, Clemens Dubslaff, Sascha Klüppelholz, Marcus Daum, Joachim Klein, Steffen Märcker, and Sascha Wunderlich. 2014. Probabilistic Model Checking and Non-standard Multi-objective Reasoning. In *FASE (LNCS)*, Vol. 8411. Springer, 1–16.
[6] Christel Baier, Joachim Klein, Sascha Klüppelholz, and Sascha Wunderlich. 2017. Maximizing the Conditional Expected Reward for Reaching the Goal. In *TACAS (LNCS)*, Vol. 10206. 269–285.
[7] Nicole Bäuerle and Jonathan Ott. 2011. Markov Decision Processes with Average-Value-at-Risk criteria. *Math. Meth. of OR* 74, 3 (2011), 361–379.
[8] Tanya Styblo Beder. 1995. VAR: Seductive but dangerous. *Financial Analysts Journal* 51, 5 (1995), 12–24.
[9] Tomás Brázdil, Václav Brozek, Krishnendu Chatterjee, Vojtech Forejt, and Antonín Kucera. 2014. Two Views on Multiple Mean-Payoff Objectives in Markov Decision Processes. *LMCS* 10, 1 (2014).
[10] Tomás Brázdil, Krishnendu Chatterjee, Vojtech Forejt, and Antonín Kucera. 2013. Trading Performance for Stability in Markov Decision Processes. In *LICS*. IEEE Computer Society, 331–340.
[11] Véronique Bruyère, Emmanuel Filiot, Mickael Randour, and Jean-François Raskin. 2017. Meet your expectations with guarantees: Beyond worst-case synthesis in quantitative games. *Inf. Comput.* 254 (2017), 259–295.
[12] John F. Canny. 1988. Some Algebraic and Geometric Computations in PSPACE. In *STOC*. ACM, 460–467.
[13] Stefano Carpin, Yinlam Chow, and Marco Pavone. 2016. Risk aversion in finite Markov Decision Processes using total cost criteria and average value at risk. In *ICRA*. IEEE, 335–342.
[14] Krishnendu Chatterjee, Vojtech Forejt, and Dominik Wojtczak. 2013. Multi-objective Discounted Reward Verification in Graphs and MDPs. In *LPAR (LNCS)*, Vol. 8312. Springer, 228–242.
[15] Krishnendu Chatterjee, Zuzana Komárková, and Jan Křetínský. 2015. Unifying Two Views on Multiple Mean-Payoff Objectives in Markov Decision Processes. In *LICS*. IEEE Computer Society, 244–256.
[16] Krishnendu Chatterjee, Zuzana Křetínská, and Jan Křetínský. 2017. Unifying Two Views on Multiple Mean-Payoff Objectives in Markov Decision Processes. *LMCS* 13, 2 (2017).
[17] Lorenzo Clemente and Jean-François Raskin. 2015. Multidimensional beyond Worst-Case and Almost-Sure Problems for Mean-Payoff Objectives. In *LICS*. IEEE Computer Society, 257–268.
[18] Costas Courcoubetis and Mihalis Yannakakis. 1995. The Complexity of Probabilistic Verification. *J. ACM* 42, 4 (1995), 857–907.
[19] Kousha Etessami, Marta Z. Kwiatkowska, Moshe Y. Vardi, and Mihalis Yannakakis. 2008. Multi-Objective Model Checking of Markov Decision Processes. *LMCS* 4, 4 (2008).
[20] J.A. Filar, D. Krass, and K.W Ross. 1995. Percentile performance criteria for limiting average Markov decision processes. *IEEE Trans. Automat. Control* 40, 1 (Jan 1995), 2–10.
[21] Jerzy A. Filar, Dmitry Krass, and Keith W. Ross. 1995. Percentile performance criteria for limiting average Markov decision processes. *IEEE Trans. Automat. Control* 40 (1995), 2–10.
[22] Vojtech Forejt, Marta Z. Kwiatkowska, Gethin Norman, David Parker, and Hongyang Qu. 2011. Quantitative Multi-objective Verification for Probabilistic Systems. In *TACAS (LNCS)*, Vol. 6605. Springer, 112–127.
[23] Hugo Gilbert, Paul Weng, and Yan Xu. 2017. Optimizing Quantiles in Preference-Based Markov Decision Processes. In *AAAI*. AAAI Press, 3569–3575.
[24] Christoph Haase and Stefan Kiefer. 2015. The Odds of Staying on Budget. In *ICALP (LNCS)*, Vol. 9135. Springer, 234–246.
[25] Christoph Haase, Stefan Kiefer, and Markus Lohrey. 2017. Computing quantiles in Markov chains with multi-dimensional costs. In *LICS*. IEEE Computer Society, 1–12.
[26] Yonghui Huang and Xianping Guo. 2016. Minimum Average Value-at-Risk for Finite Horizon Semi-Markov Decision Processes in Continuous Time. *SIAM Journal on Optimization* 26, 1 (2016), 1–28.
[27] Masayuki Kageyama, Takayuki Fujii, Koji Kanefuji, and Hiroe Tsubaki. 2011. Conditional Value-at-Risk for Random Immediate Reward Variables in Markov Decision Processes. *American J. Computational Mathematics* 1, 3 (2011), 183–188.
[28] Jan Křetínský and Tobias Meggendorfer. 2018. *Conditional Value-at-Risk for Reachability and Mean Payoff in Markov Decision Processes.* Technical Report abs/1805.xxxxx. arXiv.org.
[29] Xiaocheng Li, Huaiyang Zhong, and Margaret L. Brandeau. 2017. Quantile Markov Decision Process. *CoRR* abs/1711.05788 (2017).
[30] Christopher W. Miller and Insoon Yang. 2017. Optimal Control of Conditional Value-at-Risk in Continuous Time. *SIAM J. Control and Optimization* 55, 2 (2017), 856–884.
[31] M. L. Puterman. 1994. *Markov Decision Processes.* J. Wiley and Sons.
[32] Mickael Randour, Jean-François Raskin, and Ocan Sankur. 2017. Percentile queries in multi-dimensional Markov decision processes. *FMSD* 50, 2-3 (2017), 207–248.
[33] R. Tyrrell Rockafellar and Stanislav Uryasev. 2000. Optimization of Conditional Value-at-Risk. *Journal of Risk* 2 (2000), 21–41.
[34] R Tyrrell Rockafellar and Stanislav Uryasev. 2002. Conditional value-at-risk for general loss distributions. *Journal of banking & finance* 26, 7 (2002), 1443–1471.
[35] Michael Ummels and Christel Baier. 2013. Computing Quantiles in Markov Reward Models. In *FoSSaCS (LNCS)*, Vol. 7794. Springer, 353–368.