Trade-off Analysis Meets Probabilistic Model Checking *

Christel Baier Clemens Dubslaff Sascha Klüppelholz

Institute for Theoretical Computer Science
Technische Universität Dresden, Germany
{baier,dubslaff,klueppel}@tcs.inf.tu-dresden.de

Abstract

Probabilistic model checking (PMC) is a well-established and powerful method for the automated quantitative analysis of parallel distributed systems. Classical PMC-approaches focus on computing probabilities and expectations in Markovian models annotated with numerical values for costs and utility, such as energy and performance. Usually, the utility gained and the costs invested are dependent and a trade-off analysis is of utter interest.

In this paper, we provide an overview on various kinds of non-standard multi-objective formalisms that enable to specify and reason about the trade-off between costs and utility. In particular, we present the concepts of quantiles, conditional probabilities and expectations as well as objectives on the ratio between accumulated costs and utility. Such multi-objective properties have drawn very few attention in the context of PMC and hence, there is hardly any tool support in state-of-the-art model checkers. Furthermore, we broaden our results towards combined quantile queries, computing conditional probabilities those conditions are expressed as formulas in probabilistic computation tree logic, and the computation of ratios which can be expected on the long-run.

1. Introduction

For the quantitative analysis of probabilistic systems, various kinds of models and formal methods have been proposed in the literature. *Probabilistic model-checking* (PMC) is one very prominent example, which has been successfully applied for the quantitative analysis of hardware and software such as randomized distributed systems and even in other research domains, e.g., for the reasoning about quantitative phenomena within biological systems.

We focus here on PMC on Markovian models, which can be seen as automata annotated with probabilistic distributions and cost or reward functions modeling resource requirements. Whereas Markov chains (MCs) are purely probabilistic, Markov decision processes (MDPs) support both, nondeterministic and probabilistic choices. The typical task of PMC on a given MDP is to compute the maximal or minimal probabilities of path properties specified by some formula of linear temporal logic (LTL) [21, 41], the path-formula fragment of probabilistic computation tree logic (PCTL) or its variant PRCTL with reward-bounded temporal modalities [3, 13, 22, 31]. Algorithms for Markovian models and LTL- or PRCTL-specifications were implemented in model checkers such as PRISM [32] and MRMC [35].

In recent inter-disciplinary research projects we use PMC for the analysis of (low-level) resource-management protocols to provide insights in the energy-utility, reliability and other performance characteristics from a global and long-run perspective. With PMC we are complementing the measurement-based and simulation-based analysis conducted by our project partners. The results of the quantitative analysis guide the optimization of resource-management algorithms and can be very useful to predict the performance of management algorithms that may not have been implemented yet and aim to run on hardware that also may not yet exist neither.

Within this process the application of PMC was, however, not straightforward and we faced a number of expected challenges. Besides the state-explosion problem, which becomes even more evident in the context of (future) multi and many-core systems, additional difficulties arose to find adequate models appropriate for analyzing the properties to be investigated. A particular challenge was to find stochastic distributions modeling "realistic" workloads and complex hardware details such as cache effects. The cooperation with partners from domains such as operating systems, data bases, electrical engineering, etc. revealed further, unexpected challenges concerning the limitations of state-of-the-art PMC-techniques for the quantitative analysis. For instance, several highly relevant trade-off performance measures have been neglected by the PMC-community so far.

In this paper, we deal with PMC-methods for computing performance measures that provide insights in the trade-off between multiple cost and reward functions, such as energy and utility or system resiliency and its costs. Although there has been a recent trend for computing multi-objective properties in MDPs (see, e.g., [19, 26]), where the task is to find schedules that allow satisfying boolean combinations of constraints on probabilities and expectations, these approaches do not address the trade-off that typically exists between cost and utility: usually, the gained utility increases with the price to be payed. An example is the trade-off between energy consumption of a system and its performance. These kinds of trade-off queries can easily be formalized in terms of quantiles, conditional probabilities and (cost/utility) ratios, which are standard in statistics and mathematics, but have drawn very few attention in the context of PMC. *Quantile* queries ask, e.g., for the minimal energy required to gain a fixed minimal amount of utility. Likewise, also the question of the maximal amount of utility gained within an upper bound on the consumed energy can be formalized as a quan-

* The authors are supported by the DFG through the collaborative research centre HAEC (SFB 912), the cluster of excellence cfAED, Deutsche Telekom Stiftung, the ESF young researcher groups IMData (100098198) and SREX (100111037), the Graduiertenkolleg QuantLA (1763) the DFG/NWO-project ROCKS, and the EU-FP-7 grant MEALS (295261).

CSL-LICS 2014, July 14–18, 2014, Vienna, Austria.

http://dx.doi.org/10.1145/2603088.2603089

tile query. We presented methods for computing quantile queries in [7, 40]. Differently, *conditional probability* queries allow, e.g., to use upper or lower bounds on cost and utility measures as conditions and perform an analysis based on this given assumption. This is extremely helpful when analyzing exceptional system behavior of systems assuming that some very rare failure event occurred. Recently, it turned out that the computation of conditional probability queries can be done efficiently [10]. Quantiles and conditional probabilities can be combined very naturally. Then, the objective is to find optimal bounds on costs or utility for very specific situations that are characterized by conditions given as temporal logic constraints. *Ratio objectives* (on cost and utility) are a third important class of non-standard multi-objective queries we investigated and which allow to reason about quotients of, e.g., the accumulated cost and the accumulated utility. The importance of ratios has been realized by various other research groups. For instance, [2, 42] addressed expected ratios or [23] considered long-run ratios when the denominator has the purpose of a counter. Recent undecidability results on temporal logics extended by assertions on the accumulated values of weight functions [11, 14] impose serious limitations. These undecidability results are mainly due to the fact that this problem is closely related to MDPs with integer weights rather than non-negative rewards. Also the close relation between structures with two weight functions to two-counter machines shows that these limitations arise naturally.

Outline and Contribution This article provides an overview of recent work on the computation of quantiles, conditional probabilities and ratios of accumulated rewards in finite-state discrete Markovian models. Section 2 summarizes the preliminaries and provides a high-level introduction to the relevant concepts for MCs and MDPs. Quantiles are addressed in Section 3. We summarize our recent results presented in [7, 40] and briefly discuss the extension of quantiles towards conjunctive objectives. The transformation-based approach of [10] for the computation of conditional probabilities in Markovian models and the model-checking problem for conditional PCTL is described in Section 4. Reasoning about constraints on the ratio of the accumulated values of two reward functions and its relation to algorithmic problems for structures with an integer-valued weight function is discussed in Section 5.

2. Theoretical Foundations

Throughout the paper, we assume the reader to be familiar with concepts of standard model checking, ω-automata and temporal logics (see, e.g., [5, 20, 30]). We only provide a dense summary of the main concepts of Markov decision processes (MDPs), see, e.g., [36, 37], and the quantitative analysis of MDPs against formulas of linear temporal logic (LTL) [21, 41] and probabilistic computation tree logic (PCTL) as well as reward-based extensions thereof [3, 6, 13, 31].

Distributions. For X being a nonempty countable set, a distribution on X is a function $\mu : X \to [0,1]$ with $\sum_{x \in X} \mu(x) = 1$. We write $Distr(X)$ for the set of all distributions on X.

Markov Chains. A Markov chain $\mathcal{M}$ consists of a nonempty countable state space S and a transition probability matrix $P : S \times S \to [0,1]$, where $P(s,\cdot) \in Distr(S)$ for all states s. The associated probability space on the infinite sequences over S formalizes the intuitive notion of probabilities of (measurable) sets of sample runs in $\mathcal{M}$. If $\pi = s_0\, s_1 \ldots s_n \in S^*$, then the cylinder set of π, denoted by $Cyl(\pi)$, consists of all infinite sequences over S having π as a prefix. The cylinder sets constitute the basis of a sigma-algebra on S^ω. The probability measure $\Pr^{\mathcal{M}}_{s_{init}}$ for a given state $s_{init} \in S$ is the unique probability measure on this sigma-algebra, where $\Pr^{\mathcal{M}}_{s_{init}}(Cyl(\pi))$ equals $P(s_0,s_1) \cdot P(s_1,s_2) \cdot \ldots \cdot P(s_{n-1},s_n)$ if $s_0 = s_{init}$ and $\Pr^{\mathcal{M}}_{s_{init}}(Cyl(\pi)) = 0$ if $s_{init} \neq s_0$. The existence and uniqueness is ensured by Caratheodory's measure extension theorem.

Unless stated differently, we suppose that the state space S of a Markov chain is finite and that the transition probabilities $P(s,s')$ are rational for all $s,s' \in S$. With abuse of notations, we will use the term Markov chain as a special instance of a MDPs.

Markov Decision Processes. A Markov decision process (MDP) can be seen as a probabilistic variant of a labeled transition system, where being in a state s an action α is selected nondeterministically from a set of enabled actions, followed by a probabilistic choice of the successor state. Formally, an MDP is a tuple $\mathcal{M} = (S, Act, P, \mathsf{AP}, \mathsf{L})$, where S is a finite set of states, Act a finite set of actions, AP a finite set of atomic propositions and $\mathsf{L} : S \to 2^{\mathsf{AP}}$ a labeling function. The transition probabilities and enabled actions are specified by a function $P : S \times Act \times S \to [0,1] \cap \mathbb{Q}$ such that $\sum_{s' \in S} P(s,\alpha,s') \in \{0,1\}$ for all $s \in S$ and $\alpha \in Act$. Triples (s,α,s') where $P(s,\alpha,s') > 0$ are called transitions. We write $Act(s)$ for the set of actions that are enabled in s, i.e., $P(s,\alpha,\cdot) \in Distr(S)$ if $\alpha \in Act(s)$. For technical reasons, we require that $Act(s) \neq \varnothing$ for all states s. Markov chains can be seen as a special case of MDPs where Act is a singleton. A *pointed* MDP means a pair $(\mathcal{M}, s)$ consisting of an MDP and a distinguished initial state $s = s_{init} \in S$.

Infinite paths in $\mathcal{M}$ are infinite alternating sequences $\zeta = s_0\, \alpha_0\, s_1\, \alpha_1\, s_2\, \alpha_2 \ldots$ of states and actions built by consecutive transitions, i.e., $P(s_{i-1},\alpha_{i-1},s_i) > 0$ for all $i \geqslant 1$. If $k,\ell \in \mathbb{N}$ and $\ell \leqslant k$, then $\zeta[\ell \ldots k]$ denotes the path fragment $s_\ell\, \alpha_\ell \ldots \alpha_{k-1}\, s_k$. Hence, $\zeta[0 \ldots k] = \mathit{pref}(\zeta,k)$ is the prefix of ζ consisting of the first k transitions and $\zeta[k \ldots k] = \zeta[k] = s_k$ denotes the $(k{+}1)$-st state in ζ. The trace of ζ is the infinite word

$$\mathsf{trace}(\zeta) \;=\; \mathsf{L}(s_0)\,\mathsf{L}(s_1)\,\mathsf{L}(s_2)\ldots \;\in\; \left(2^{\mathsf{AP}}\right)^{\omega}$$

that is obtained by the projection to the atomic propositions the states in ζ are labeled with. *Finite paths* are nonempty finite prefixes of infinite paths. The length $|\pi|$ of a finite path π denotes the number of transitions in π. The notations $\mathsf{trace}(\pi)$, $\pi[\ell \ldots k]$, $\pi[k]$ for $0 \leqslant \ell \leqslant k \leqslant |\pi|$ are defined as for infinite paths, and $\mathit{last}(\pi) = \pi[|\pi|]$ denotes the last state of π. $\mathit{FinPaths}$ and $\mathit{InfPaths}$ stand for the set of all finite resp. infinite paths, whereas $\mathit{FinPaths}(s)$ and $\mathit{InfPaths}(s)$ denote the corresponding sets of paths starting in state s.

Schedulers and Probability Measure. Reasoning about probabilities for path properties in an MDP $\mathcal{M}$ requires the selection of an initial state and the resolution of the nondeterministic choices between the possible transitions. This is formalized via *schedulers*, which take as input the history, formalized by a finite path, and select an action to be executed next. Formally, a (randomized history-dependent) scheduler for $\mathcal{M}$ is a function $\mathfrak{S} : \mathit{FinPaths} \to Distr(Act)$ such that for all actions $\alpha \in Act$ and finite paths π holds $\alpha \in Act(\mathit{last}(\pi))$ if $\mathfrak{S}(\pi)(\alpha) > 0$. An $\mathfrak{S}$-path is an infinite path $\zeta = s_0\, \alpha_0\, s_1\, \alpha_1\, s_2\, \alpha_2 \ldots$ in $\mathcal{M}$ that can arise following $\mathfrak{S}$'s decisions, i.e., $\mathfrak{S}(\zeta[0 \ldots k])(\alpha_k) > 0$ for all $k \in \mathbb{N}$. Scheduler $\mathfrak{S}$ is said to be *deterministic* if for each finite path π there is an action α with $\mathfrak{S}(\pi)(\alpha) = 1$, in which case we regard $\mathfrak{S}$ as a function $\mathfrak{S} : \mathit{FinPaths} \to Act$. *Finite-memory* schedulers operate with a finite-state transducer to store the relevant information on the history. A special case are *memoryless* schedulers whose decisions only depend on the last state. Memoryless schedulers can be formalized as functions $\mathfrak{S} : S \to Distr(Act)$ (randomized memoryless schedulers) or $\mathfrak{S} : S \to Act$ (deterministic memoryless schedulers).

The behavior of a pointed MDP $(\mathcal{M}, s)$ under $\mathfrak{S}$ is purely probabilistic and can be formalized by a tree-like infinite-state

Markov chain. This yields the probability for a measurable path property φ (e.g., specified by an LTL or reward-bounded formula, see below) under $\mathfrak{S}$ starting in state s:

$$\mathrm{Pr}_s^{\mathfrak{S}}(\varphi) \quad = \quad \mathrm{Pr}_s^{\mathfrak{S}}\big\{\, \zeta \in \mathit{InfPaths}(s) \,:\, \zeta \models \varphi \,\big\}$$

For a worst- or best-case analysis of a system modeled by a pointed MDP $(\mathcal{M}, s)$, one ranges over all schedulers (i.e., all possible resolutions of the nondeterminism) and considers the maximal or minimal probabilities for satisfying φ:

$$\mathrm{Pr}_s^{\min}(\varphi) = \inf_{\mathfrak{S}} \mathrm{Pr}_s^{\mathfrak{S}}(\varphi) \quad \text{and} \quad \mathrm{Pr}_s^{\max}(\varphi) = \sup_{\mathfrak{S}} \mathrm{Pr}_s^{\mathfrak{S}}(\varphi)$$

For many relevant path properties φ, e.g., all ω-regular properties, optimal finite-memory schedulers that maximize or minimize the probability to satisfy φ do exist. Occasionally, we use $\mathcal{M}$ as an additional subscript and write $\mathrm{Pr}_{\mathcal{M},s}^{\max}(\varphi)$ or $\mathrm{Pr}_{\mathcal{M},s}^{\max}(\varphi)$.

End Components. An MDP-analogue to ergodic subsets of finite-state Markov chains [22] are provided by strongly connected sub-MDPs called *end components*. Formally, an end component of $\mathcal{M}$ is a pair $\mathfrak{E} = (T, \mathfrak{A})$ consisting of a nonempty subset T of S and a function $\mathfrak{A} : T \to 2^{Act} \backslash \{\varnothing\}$ such that:

- $\varnothing \neq \mathfrak{A}(t) \subseteq Act(t)$ for all states $t \in T$,
- if $t \in T$, $\alpha \in \mathfrak{A}(t)$ and $P(t, \alpha, t') > 0$ then $t' \in T$, and
- the underlying directed graph is strongly connected.

Here, the underlying graph of $(T, \mathfrak{A})$ has the node-set T and an edge (t, t') iff there exists $\alpha \in \mathfrak{A}(t)$ with $P(t, \alpha, t') > 0$. It is well-known [22] that under each scheduler $\mathfrak{S}$, the limit of almost all infinite paths constitutes an end component. In particular, the limit of an infinite path ζ is the pair $\mathrm{Limit}(\zeta) = (T, \mathfrak{A})$, where $T = \inf(\zeta)$ is the set of states t that appear infinitely often in ζ and for which $\mathfrak{A}(t)$ is the set of actions that are taken infinitely often in ζ from t. The computation of maximal end components, i.e., end components that are not contained in any other end component, can be computed using an iterative approach for computing strongly connected components in subgraphs [18, 21, 22].

Linear Temporal Logic. The linear temporal logic LTL extends propositional logic over AP by the temporal modalities $\bigcirc$ (next) and U (until). Other temporal modalities can then be derived, e.g., $\Diamond\varphi = \mathit{true}\mathrm{U}\varphi$ (eventually), $\Box\varphi = \neg\Diamond\neg\varphi$ (always), $\varphi_1 \mathrm{R} \varphi_2 = \neg(\neg\varphi_1 \mathrm{U} \neg\varphi_2)$ (release). LTL formulas are interpreted over pairs (w, k) consisting of an infinite word $\mathsf{w} = \mathsf{A}_0\, \mathsf{A}_1\, \mathsf{A}_2 \in (2^{\mathsf{AP}})^\omega$ and a word position $k \in \mathbb{N}$. For instance, if $\mathsf{a} \in \mathsf{AP}$ then $(\mathsf{w}, k) \models \mathsf{a}$ iff $\mathsf{a} \in \mathsf{A}_k$, while the semantics of the temporal modalities $\bigcirc$ and U is given by: $(\mathsf{w}, k) \models \bigcirc\varphi$ iff $(\mathsf{w}, k{+}1) \models \varphi$ and $(\mathsf{w}, k) \models \varphi_1 \mathrm{U} \varphi_2$ iff there exists an $\ell \in \mathbb{N}$, where $\ell \geqslant k$, $(\mathsf{w}, \ell) \models \varphi_2$ and $(\mathsf{w}, i) \models \varphi_1$ for all $k \leqslant i < \ell$. An infinite word $\mathsf{w} \in (2^{\mathsf{AP}})^\omega$ is said to be a model for φ iff $(\mathsf{w}, 0) \models \varphi$. If φ is an LTL formula and $\zeta \in \mathit{InfPaths}$, then $\zeta \models \varphi$ if $\mathsf{trace}(\zeta)$ is a model for φ. $\mathrm{Pr}_s^{\min}(\varphi)$ and $\mathrm{Pr}_s^{\max}(\varphi)$ refer to the minimal or maximal probabilities for the set of infinite paths satisfying φ, ranging over all schedulers for $\mathcal{M}$.

To specify properties for MDPs, we often use sets of states or single states as atomic propositions with the obvious meaning.

Figure 1 sketches the main steps of the automata-based approach for probabilistic model checking (PMC) MDPs against LTL-specifications. Maxima and minima are taken over all potential resolutions of the nondeterminism, formalized by schedulers. We suppose here that the formula φ describes the undesired behaviors, i.e., the behaviors where the requirement does not hold. In this case, the maximal probability for satisfying φ and corresponding schedulers provide insights in the worst-case scenarios.

The idea is to apply at first known algorithms that transform the given LTL-formula into a deterministic automaton $\mathcal{A}$ over infinite words (see [30]) and then to compute the maximal probabilities for

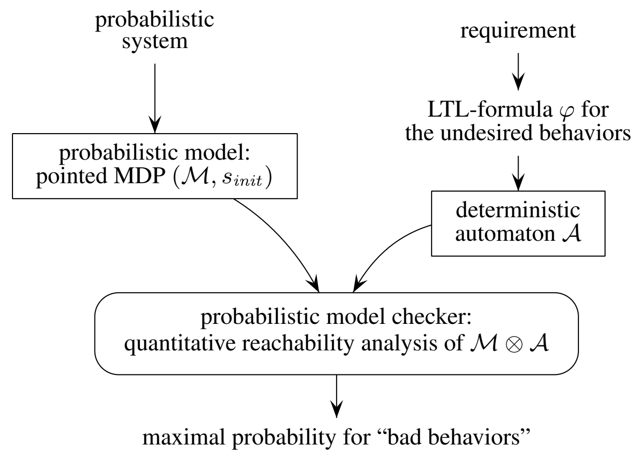


Figure 1. Automata-based PMC for LTL-specifications

the paths satisfying $\mathcal{A}$'s acceptance condition in the product-MDP $\mathcal{M} \otimes \mathcal{A}$. The latter agrees with the maximal probability to reach an end component of $\mathcal{M} \otimes \mathcal{A}$ that meets the acceptance condition of $\mathcal{A}$. Thus, the task to compute $\mathrm{Pr}_s^{\max}(\varphi)$ reduces to a probabilistic reachability problem in the product-MDP and is solvable by linear-programming techniques (as sketched for the PCTL model checking procedure, see below). A worst-case analysis as in Fig. 1 is adequate if the choices between the nondeterministic alternatives in the given MDP are uncontrollable, e.g., if they represent the possible interactions with an unknown or unpredictable environment. If the given LTL-formula φ formalizes the desired behaviors, the computation of the maximal probability for φ can be seen as a best-case analysis. Then, a scheduler maximizing the probability for φ serves as an optimal controller for the objective φ.

Probabilistic Computation Tree Logic. A probabilistic variant of CTL that replaces the path quantifiers $\exists$ and $\forall$ with a probabilistic operator P that serves to assert bounds for the worst- or best-case probabilities for simple path properties is given by probabilistic computation tree logic (PCTL) [13, 31]. It can be used in combination with existential or universal quantification over schedulers. PCTL state formulas are Boolean combinations of atomic propositions $\mathsf{a}, \mathsf{b}, \mathsf{c}, \ldots \in \mathsf{AP}$ and formulas of the type $\exists\mathsf{P}_{\bowtie q}(\varphi)$ or $\forall\mathsf{P}_{\bowtie q}(\varphi)$, where $\bowtie$ is a comparison operator $\leqslant$, $<$, $\geqslant$ or $>$, $q \in [0, 1] \cap \mathbb{Q}$ a probability bound and φ a PCTL path formula, i.e., a formula of the form $\bigcirc\Phi$ or $\Phi \mathrm{U} \Psi$ with PCTL state formulas Φ and Ψ. As in LTL, $\bigcirc$ and U stand for the temporal modalities "next" and "until". We refer to the terms $\mathsf{P}_{\bowtie q}(\varphi)$ as *probability constraints*. Thus, PCTL state formulas are Boolean combination of existentially or universally quantified probability constraints.

The interpretation of PCTL path and state formulas over the infinite paths resp. the states of an MDP $\mathcal{M}$ is defined by structural induction. Satisfaction of a path formula or propositional logical state formula is satisfied as for CTL, e.g., $\zeta \models \bigcirc\Phi$ iff $\zeta[1] \models \Phi$ and $s \models \neg\Phi$ iff $s \not\models \Phi$. The semantics of the probabilistic operator is given by $s \models \forall\mathsf{P}_{\bowtie q}(\varphi)$ iff $\mathrm{Pr}_s^{\mathfrak{S}}(\varphi) \bowtie q$ for all schedulers $\mathfrak{S}$, where (as before) φ is identified with the set of infinite paths ζ such that $\zeta \models \varphi$. Similarly, $s \models \exists\mathsf{P}_{\bowtie q}(\varphi)$ iff $\mathrm{Pr}_s^{\mathfrak{S}}(\varphi) \bowtie q$ for some scheduler $\mathfrak{S}$. Other temporal modalities like $\Diamond$ (eventually), $\Box$ (always) and R (release) can be derived in a similar fashion. For instance, $\Diamond\Phi \stackrel{\text{def}}{=} \mathit{true}\mathrm{U}\Phi$ and $\forall\mathsf{P}_{<q}(\Box\Phi) \stackrel{\text{def}}{=} \neg\exists\mathsf{P}_{\leqslant 1-q}(\Diamond\neg\Phi) \equiv \neg\exists\mathsf{P}_{\geqslant q}(\Box\Phi)$ where $\equiv$ denotes the equivalence of formulas. For a PCTL state formula Φ and a pointed MDP $(\mathcal{M}, s_{init})$, the model-checking problem for PCTL amounts of checking whether $s_{init} \models \Phi$. It is solvable by an inductive approach, where the satisfaction

sets $Sat(\Psi) = \{s \in S : s \models \Psi\}$ of all state sub-formulas Ψ of Φ are computed inductively. Obviously, then $s_{init} \models \Phi$ iff $s_{init} \in Sat(\Phi)$. The treatment of the propositional logic fragment is obvious as we have $Sat(\mathsf{a}) = \{s \in S : \mathsf{a} \in \mathsf{L}(s)\}$, $Sat(\neg\Psi) = S \backslash Sat(\Psi)$ and $Sat(\Psi_1 \wedge \Psi_2) = Sat(\Psi_1) \cap Sat(\Psi_2)$. For the probabilistic operator, the computation of maximal or minimal probabilities for PCTL path formulas is required:

$$\begin{aligned} Sat(\forall \mathsf{P}_{\trianglelefteq q}(\varphi)) &= \{ s \in S : \Pr_s^{\max}(\varphi) \trianglelefteq q \} \\ Sat(\forall \mathsf{P}_{\trianglerighteq q}(\varphi)) &= \{ s \in S : \Pr_s^{\min}(\varphi) \trianglerighteq q \}, \end{aligned}$$

where $\trianglelefteq \in \{\leqslant, <\}$ and $\trianglerighteq \in \{\geqslant, >\}$. The satisfaction sets for state formulas with existential scheduler quantification are obtained in the same way, except that we replace $\Pr_s^{\max}$ with $\Pr_s^{\min}$ and vice versa. Let us briefly explain the treatment of upper probability bounds and universal scheduler quantification. If $\varphi = \bigcirc\Psi$, then $\Pr_s^{\max}(\varphi) = \max\{P(s, \alpha, Sat(\Psi)) : \alpha \in Act(s)\}$. In case of $\varphi = \Psi_1 \mathrm{U} \Psi_2$, where $A = Sat(\Psi_1)$ and $B = Sat(\Psi_2)$, the vector $(p_s)_{s \in S}$ with $p_s = \Pr_s^{\max}(A \mathrm{U} B)$ constitutes the unique solution of the following linear program with variables x_s for $s \in S$:

minimize $\sum_{s \in S_?} x_s$ subject to

$$\begin{aligned} x_s &= 0 && \text{if } s \not\models \exists(A \mathrm{U} B) \\ x_s &= 1 && \text{if } \Pr_s^{\max}(A \mathrm{U} B) = 1 \\ x_s &\geqslant \sum_{s' \in S} P(s, \alpha, s') \cdot x_{s'} && \text{if } s \in S_?, \alpha \in Act(s) \end{aligned}$$

Here, $S_?$ denotes the set all states s with $0 < \Pr_s^{\max}(A \mathrm{U} B) < 1$. Containment of a state s in $S_?$ is efficiently decidable, since $s \not\models \exists(A \mathrm{U} B)$ and $\Pr_s^{\max}(A \mathrm{U} B) = 1$ can be also efficiently decided using standard graph algorithms.

Since the concept of schedulers is irrelevant for Markov chains, PCTL interpreted over the states and paths in Markov chains does not require scheduler quantifiers. That is, PCTL formulas can then be written as Boolean combinations of probability constraints. The above linear program for the computation of the probabilities for $A \mathrm{U} B$ can then be rephrased as a linear equation system.

Weight and Reward Functions. A *weight function* for $\mathcal{M}$ is a function of the form $wgt : S \times Act \to \mathbb{Z}$ that assigns an integer to all state-action pairs where $wgt(s, \alpha) = 0$ if $\alpha \notin Act(s)$. When wgt is non-negative, i.e., $wgt(s, s') \geqslant 0$ for all states s, s', then we refer to wgt as a *reward function*. We say wgt is *positive* if $wgt(s, \alpha) > 0$ for all state-action pairs (s, α) where α is enabled in s. Occasionally, we also consider weight functions with rational values, which we call *rational-valued* weight functions. The *accumulated weight* of finite paths is defined by:

$$wgt(s_0\, \alpha_0\, s_1\, \alpha_1 \ldots \alpha_{n-1}\, s_n) = \sum_{0 \leqslant i < n} wgt(s_i, \alpha_i)$$

Expected Accumulated Reward. Besides the computation of extremal probabilities for path properties, probabilistic model-checking techniques are also available for reasoning about expectations of random variables. We consider here the extremal expected accumulated rewards for some given reward function $rew : S \times Act \to \mathbb{N}$ until some event occurs.

The first instance concerns reachability events formalized by a state predicate $goal$ (i.e., Boolean combination of atomic propositions) that induces the random variable $f[rew \!\downarrow\! goal] : InfPaths \to \mathbb{N} \cup \{\infty\}$ with $f[rew \downarrow goal](\zeta) = rew(\zeta[0 \ldots k])$ and $k = \min\{\ell \in \mathbb{N} : \zeta[\ell] \models goal\}$. If $\zeta \not\models \Diamond goal$, then $f[rew \downarrow goal](\zeta) = \infty$. Assuming that $\Pr_s^{\min}(\Diamond goal) = 1$, the expectation of $f[rew \!\downarrow\! goal]$ exists and is finite for all schedulers $\mathfrak{S}$ and so are the maximal and minimal expected rewards:

$$\begin{aligned} \mathrm{E}_s^{\max}[rew \!\downarrow\! goal] &= \max_{\mathfrak{S}} \mathrm{E}_s^{\mathfrak{S}}[rew \!\downarrow\! goal] \\ \mathrm{E}_s^{\min}[rew \!\downarrow\! goal] &= \min_{\mathfrak{S}} \mathrm{E}_s^{\mathfrak{S}}[rew \!\downarrow\! goal] \end{aligned}$$

The extremal expectations are computable using linear-programming techniques and the existence of corresponding optimal deterministic memoryless schedulers can be guaranteed [24, 37]. To study the trade-off between two reward functions rew_1 and rew_2, we deal with the random variable $f[rew_1 \!\uparrow\! (rew_2 \!\leqslant\! r)] : InfPaths \to \mathbb{N} \cup \{\infty\}$ that assigns to each infinite path ζ the accumulated reward with respect to rew as long as the accumulated reward with respect to rew_2 is bounded by r. More precisely, $f[rew_1 \!\uparrow\! (rew_2 \!\leqslant\! r)](\zeta)$ equals $\sup\{rew_1(\zeta[0 \ldots k]) : rew_2(\zeta[0 \ldots k]) \leqslant r\}$. Then, $\mathrm{E}_s^{\max}[rew_1 \!\uparrow\! (rew_2 \!\leqslant\! r)]$ and $\mathrm{E}_s^{\min}[rew_1 \!\uparrow\! (rew_2 \!\leqslant\! r)]$ are defined as the maximal resp. minimal expectation of $f[rew_1 \!\uparrow\! (rew_2 \!\leqslant\! r)]$ when ranging over all schedulers and assuming that almost all infinite paths have a finite prefix π with $rew_2(\pi) > r$. For example, if $rew_1 = utility$ formalizes the gained utility and $rew_2 = energy$ the consumed energy, then $\mathrm{E}_s^{\max}[utility \uparrow (energy \leqslant e)]$ stands for the maximal expected utility that can be achieved under some scheduler when only the energy budget e is available. Note that $f[rew \!\downarrow\! goal]$ equals $f[rew_1 \!\uparrow\! (rew_2 \!\leqslant\! 0)]$ when $rew_1 = rew$ and rew_2 is a fresh reward function with $rew_2(s, \alpha) = 1$ if $s \models goal$ and $\alpha \in Act(s)$ and $rew_2(s, \alpha) = 0$ in all other cases.

PCTL with Rewards (PRCTL). For reasoning about reward-based properties, PCTL can be extended to also support statements about several reward functions in MDPs. In case of a single reward function rew, PRCTL extends PCTL by state formulas of the form $\forall \mathsf{P}_{\bowtie q}(\varphi[r])$ and $\forall \mathsf{E}_{\asymp \vartheta}(rew \downarrow \Phi)$ and analogous formulas with existential scheduler quantification. Here, $q \in [0, 1] \cap \mathbb{Q}$ is a probability bound and ϑ is a non-negative rational threshold for the extremal expected reward. Furthermore, $\varphi[r]$ is a reward-bounded path property depending on a reward bound $r \in \mathbb{N}$. In this paper, we only consider reward-bounded modalities in the form $\Phi \mathrm{U}^{\asymp r} \Psi$, where $r \in \mathbb{N}$ is a reward bound, $\asymp$ is a comparison operator in $\{\leqslant, <, \geqslant, >, =\}$ and Φ and Ψ are PRCTL state formulas. Given an MDP with a reward function rew and an infinite path ζ in $\mathcal{M}$, then $\zeta \models \Phi \mathrm{U}^{\asymp r} \Psi$ iff there is some $k \in \mathbb{N}$ with $\zeta[k] \models \Psi$, $\zeta[i] \models \Phi$ for all $0 \leqslant i < k$, and $rew(\zeta[0 \ldots k]) \asymp r$. The formula $\forall \mathsf{E}_{\leqslant r}(rew \downarrow \Phi)$ asserts that under each scheduler the expected accumulated reward until reaching a state s with $s \models \Phi$ is at most r. (To ensure well-definedness of the semantics we require that $\Pr_s^{\min}(\Diamond Sat(\Phi)) = 1$.) The meaning of expectation operator with lower bounds is analogous.

Likewise, we can attach reward bounds to the until operator for the accumulated reward with respect to several reward functions. In our examples, we only consider the case of MDPs with two reward functions $rew_1 = utility$ with lower reward bounds (written as lower subscripts) and $rew_2 = energy$ with upper reward bounds (written as upper subscripts). Thus, $\mathrm{U}_{\geqslant u}^{\leqslant e}$ denotes the until operator with upper energy bound e and lower utility bound u. This allows for reasoning about the trade-off between two reward functions, such as energy and utility. Furthermore, we can deal with universal or existential expectation constraints, such as $\forall \mathsf{E}_{\asymp \vartheta}(utility \uparrow (energy \leqslant e))$ stating that the expected utility accumulated as long as the consumed energy is at most e meets the bound $\asymp \vartheta$ for each scheduler.

3. Quantiles

Quantiles are well-established in statistics (see, e.g., [38]), where they are used to reason about the cumulative distribution function of a random variable R. If $q \in]0, 1[$, then the q-quantile is the maximal value r such that the probability for the event $R > r$ is at least q. Although quantiles can provide very useful insights in the interplay of various cost functions and other system properties, they have been barely obtained attention in the model-checking community. Quantiles adapted to Markovian models serve to formalize optimization problems where the task is to maximize or minimize some reward parameter r subject to a parametrized probability or expectation constraint [7, 40].

Quantiles Under Probability Constraints. Let us start with a simple example. Suppose we are given a pointed Markov chain with two reward functions $rew_1 = utility$ and $rew_2 = energy$, modeling an energy-aware job scheduling protocol. The utility reward function might stand for the profit in terms of money obtained for the successfully completed tasks without violating the service level agreement (SLA), whereas the energy reward function formalizes the consumed energy. When $goal$ represents some target, e.g., the set of all states where each job is either completed or discarded if its deadline has been expired, the double reward-bounded property

$$\varphi_{e,u} \;=\; \Diamond^{\leqslant e}_{\geqslant u}\, goal$$

asserts that the target will be reached while the consumed energy is at most e and the gained utility is at least u. Formally, $\zeta \models \varphi_{e,u}$ iff there is some $k \in \mathbb{N}$ with $utility(\zeta[0\ldots k]) \geqslant u$ and $energy(\zeta[0\ldots k]) \leqslant e$. Assuming some lower bound u on the "acceptable" utility, then the probability for $\varphi_{e,u}$ is increasing in e and may define the quantile formalizing the minimal amount $e_{\min}$ of energy required to guarantee that the utility is at least u with probability > 0.8. Likewise, when we fix the energy budget e, then the probability for $\varphi_{e,u}$ is decreasing in u and a quantile can be used to define the maximal utility value $u_{\max}$ that can be achieved for the energy budget e with probability > 0.8. This quantile example is illustrated in Figure 2.

Suppose now that the given model is an MDP and that the accumulated rewards for all but one reward function are encoded using program variables that serve as reward counters. For instance, with a fixed energy budget e, we model the energy consumption by a program variable, in which case the formula $\varphi_{e,u}$ can be replaced with $\Diamond_{\geqslant u}(goal \wedge (energy \leqslant e))$. We now may require that the probability constraint $\mathsf{P}_{>0.8}(\varphi_{e,u})$ holds for all schedulers or for some scheduler. This leads to the general definition of quantiles in MDPs with respect to probabilistic constraints for until properties with upper or lower reward bounds and lower probability bounds (where $\trianglerighteq \in \{\geqslant, >\}$ and $q \in [0,1] \cap \mathbb{Q}$):

universal quantiles:

$$\min\ \{\, r \in \mathbb{N} : \Pr^{\min}(A \mathrm{U}^{\leqslant r} B) \trianglerighteq q \,\}$$
$$\max\ \{\, r \in \mathbb{N} : \Pr^{\min}(A \mathrm{U}_{\geqslant r} B) \trianglerighteq q \,\}$$

existential quantiles:

$$\min\ \{\, r \in \mathbb{N} : \Pr^{\max}(A \mathrm{U}^{\leqslant r} B) \trianglerighteq q \,\}$$
$$\max\ \{\, r \in \mathbb{N} : \Pr^{\max}(A \mathrm{U}_{\geqslant r} B) \trianglerighteq q \,\}$$

Quantiles for upper-reward bounded until properties with *qualitative* probability constraints (i.e., probability bounds $= 1$, < 1, $= 0$ or > 0), can be computed in polynomial time using a greedy method that shares some ideas of Dijkstra's shortest-path algorithm [40] and relies on expansion laws for qualitative PRCTL properties such as:

$$\forall\mathsf{P}_{=1}(A \mathrm{U}^{\leqslant r} B) \;\equiv\; B \vee \big(A \wedge \forall\mathsf{P}_{=1}(A \mathrm{U}^{=0} \forall\mathsf{P}_{=1}(A \mathrm{U}^{\leqslant r} B))\big)$$

For probability bounds $q \in\,]0,1[$, the schema for computing quantiles is as follows. We explain here the case to find the minimal reward bound $r{=}r_s$ such that the probabilistic constraint $\mathsf{P}_{<q}(\Diamond_{\geqslant r} B)$ holds for all schedulers, where $0 < q < 1$ and B is a set of goal states. That is, $r_s = \min\{r \in \mathbb{N} : \Pr^{\max}_s(\Diamond_{\geqslant r} B) < q\}$. Note that then, $r_s{-}1$ is the maximal reward bound such that the probabilistic constraint $\mathsf{P}_{\geqslant q}(\Diamond_{\geqslant r} B)$ holds for some scheduler.

1. We first apply standard PMC-techniques for LTL to compute $p_s = \Pr^{\max}_s(\Diamond(C \wedge \Diamond B))$ for all states s, where C is the set of all states that belong to some end component containing at least one state-action pair with positive reward. With $X = \{s \in S : p_s \geqslant q\}$ we have $r_s = \infty$ iff $s \in X$.
2. If $S \neq X$, then for $r = 0, 1, 2, \ldots$ we compute the values $p_{s,r} = \Pr^{\max}_s(\Diamond_{\geqslant r} B)$ for all $s \in S$ and proceed with step 3 as soon as $p_{s,r} < q$ for all states $s \in S \setminus X$.
3. For each $s \in S \setminus X$, return $r_s = \min\{r \in \mathbb{N} : p_{s,r} < q\}$.

The computation of the values $p_{s,r}$ in step 2 can be carried out using linear-programming techniques, based on the fact that $p_{s,0} = \Pr^{\max}_s(\Diamond B)$ and that the vector $(p_{s,r})_{s \in S}$ for $r > 0$ is the unique solution of the following linear program (see [7] for further details):

minimize $\sum_{s \in S_?} x_s$ subject to

$$x_s \;=\; 0 \quad \text{if } s \not\models \exists\Diamond B$$
$$x_s \;\geqslant\; 0 \quad \text{if } s \models \exists\Diamond B$$

if $s \models \exists\Diamond B$, $\alpha \in Act(s)$ and $\ell = rew(s,\alpha)$, then:

$$x_s \;\geqslant\; \sum_{s' \in S} P(s,\alpha,s') \cdot x_{s'} \qquad \text{if } \ell = 0$$
$$x_s \;\geqslant\; \sum_{s' \in S} P(s,\alpha,s') \cdot p_{s',r-\ell} \qquad \text{if } \ell > 0$$

Quantiles Under Expectation Constraints. Expectation quantiles serve to formalize optimization tasks, which aim towards minimizing or maximizing the expected value of a parametrized random variable. For instance, considering an MDP with two reward function $rew_1 = utility$ and $rew_2 = energy$ and a given utility value u, we might ask for a scheduler that minimizes the expected energy consumption required to ensure that the expected gained utility exceeds u. This corresponds to the query "synthesize a scheduler $\mathfrak{S}$ such that $\mathrm{E}^{\mathfrak{S}}_s[utility{\uparrow}(energy \leqslant e_{\min})] > u$", where

$$e_{\min} \;=\; \min\big\{\, e \in \mathbb{N} : \mathrm{E}^{\max}_s[utility{\uparrow}(energy{\leqslant}e)] > u \,\big\}.$$

The computation of expectation quantiles can follow an analogous approach as for probability quantiles. The idea is to first identify the states where the expectation quantile is infinite and then iteratively compute the values $u_{s,e} = \mathrm{E}^{\max}_s[utility \uparrow (energy{\leqslant}e)]$ for $e = 0, 1, \ldots$ until $u_{s,e} > u$. Again, the latter step relies on the fact that the vector $(u_{s,e})_{s \in S}$ is the unique solution of the linear program:

minimize $\sum_{s \in S_?} x_s$ subject to $x_s \geqslant 0$ and

if $s \in S$, $\alpha \in Act(s)$ and $rew(s,\alpha) = 0$ then:

$$x_s \;\geqslant\; utility(s,\alpha) + \sum_{s' \in S} P(s,\alpha,s') \cdot x_{s'}$$

if $s \in S$, $\alpha \in Act(s)$ and $0 < \ell = rew(s,\alpha) \leqslant e$ then:

$$x_s \;\geqslant\; utility(s,\alpha) + \sum_{s' \in S} P(s,\alpha,s') \cdot u_{s',e-\ell}$$

Quantiles Under Conjunctive Constraints. So far, we considered only quantiles that are defined as minima or maxima over reward bounds such that a single probability or expectation constraint holds. However, the linear-programming approach sketched above is also applicable to compute quantiles with a conjunction of universal probability or expectation constraints. An example for such a quantile is the minimal energy budget e required to ensure that

$$\forall\mathfrak{S}.\ \Pr^{\mathfrak{S}}_s(\Diamond^{\leqslant e} goal) > 0.8 \wedge \mathrm{E}^{\mathfrak{S}}_s[utility{\uparrow}(energy \leqslant e)] > u,$$

where u is a fixed non-negative rational utility threshold. This is equivalent to the requirement that $s \models \forall\mathsf{P}_{>0.8}(\Diamond^{\leqslant e} goal) \wedge \forall\mathsf{E}_{>u}(utility{\uparrow}(energy \leqslant e))$. For computing the quantile

$$r_s \;=\; \min\big\{\, r \in \mathbb{N} : s \models \forall(C_1[r] \wedge \ldots \wedge C_k[r]) \,\big\},$$

$e_{\min} = \min\{e \in \mathbb{N} : \Pr_s(\Diamond^{\leqslant e}_{\geqslant u}\ goal) > 0.8\}$ $\quad$ $u_{\max} = \max\{u \in \mathbb{N} : \Pr_s(\Diamond^{\leqslant e}_{\geqslant u}\ goal) > 0.8\}$

Figure 2. Quantiles for increasing and decreasing properties

where $C_1, \ldots, C_k$ are probability or expectation constraints with reward parameter r, we can treat the probability and expectation constraints separately to compute the sets $R_i = \{r \in \mathbb{N} : s \models \forall C_i[r]\}$. Then, r_s is the minimal value in $R_1 \cap \ldots \cap R_k$. Note that the universal conjunctive constraint $\forall(C_1[r] \wedge \ldots \wedge C_k[r])$ is equivalent to the PRCTL formula $\forall C_1[r] \wedge \ldots \wedge \forall C_k[r]$.

The analogous problem for existential rather than universal scheduler quantification asks to find a sharp reward bound $r_s = \min\{r \in \mathbb{N} : s \models \exists(C_1[r] \wedge \ldots \wedge C_k[r])\}$ such that a conjunction of probability and/or expectation constraints holds under some scheduler. For example, the task might be to find a scheduler $\mathfrak{S}$ that yields a witness for

$$\exists \mathfrak{S}.\ \Pr^{\mathfrak{S}}_s(\Diamond^{\leqslant e} goal) > 0.8 \wedge \mathrm{E}^{\mathfrak{S}}_s[\mathit{utility}{\uparrow}(\mathit{energy} \leqslant e)] > u$$

The computation of such quantiles can employ linear-programming techniques that have been suggested for standard multi-objective queries [26, 27]. In this case, the task is to find a scheduler such that a conjunction of probability constraints holds for ω-regular properties and expectation constraints for the total accumulated reward until reaching a goal. This approach can be adapted to the setting of constraints with reward parameters. We briefly explain the case where the $C_j[r]$'s are reward-bounded reachability constraints of the form $\mathsf{P}_{>q_j}(\Diamond^{\leqslant r} B_j)$ and assume that $B_1, \ldots, B_k$ are pairwise disjoint sets of states with $Act(s) = \{\tau\}$ and $P(s, \tau, \mathit{trap}) = 1$ for all $s \in B_1 \cup \cdots \cup B_k$ and some trap state trap. For a distinguished initial state s_{init}, the task then amounts of computing

$$\min\{r \in \mathbb{N} : s_{init} \models \exists(\mathsf{P}_{>q_1}(\Diamond^{\leqslant r} B_1) \wedge \ldots \wedge \mathsf{P}_{>q_k}(\Diamond^{\leqslant r} B_k))\}$$

To check the existence of such a value r, we can rely on the linear-programming techniques presented in [26] and check the existence of a scheduler $\mathfrak{S}$ such that $\Pr^{\mathfrak{S}}_{s_{init}}(\Diamond B_j) > q_j$ for all $j = 1, \ldots, k$. If so, we then consider $r = 0, 1, 2, \ldots$ until for some scheduler $\mathfrak{S}$ we have $\Pr^{\mathfrak{S}}_{s_{init}}(\Diamond^{\leqslant r} B_j) > q_j$ for $j = 1, \ldots, k$. To check the latter for given $r \in \mathbb{N}$, we use a linear program with variables $y_{s,\alpha,i}$ with $s \in S \backslash \{\mathit{trap}\}$, $\alpha \in Act(s)$ and $i \in \{0, 1, \ldots, r\}$. These variables represent the expected number of times that action α will be taken in state s when the accumulated reward is i under some randomized scheduler. The decisions of this randomized scheduler may depend on the current state s and the reward i that has been accumulated in the past. The linear constraints consist of the requirements that the variables $y_{s,\alpha,i}$ are non-negative and satisfy the flow equation

$$\mathit{init}(s) + \sum_{(s',\beta) \in X(i)} P(s', \beta, s) \cdot y_{s',\beta,i-rew(s',\beta)} = \sum_{\alpha \in Act(s)} y_{s,\alpha,i},$$

where $\mathit{init}(s) = 1$ if $s = s_{init}$ and $\mathit{init}(s) = 0$ otherwise and $X(i)$ denotes the set of all state-action pairs (s', β) with $s' \in S$, $\beta \in Act(s')$, and $rew(s', \beta) \leqslant i$. For $s \in B_j$ we have the additional constraint $y_{s,\alpha,r} \geqslant q_j$. The minimal values $y^*_{s,\alpha,i}$ satisfying these linear constraints indeed encode a randomized finite-memory scheduler $\mathfrak{S}$ with $\Pr^{\mathfrak{S}}_{s_{init}}(\Diamond B_j) > q_j$ for $j = 1, \ldots, k$. If the current state s and the accumulated reward is i, then $\mathfrak{S}$ picks action α with probability $y^*_{s,\alpha,i}/z_{s,i}$, provided that $y_{s,\alpha,i} > 0$ and $z_{s,i} = \sum_{\alpha \in Act(s)} y^*_{s,\alpha,i}$.

4. Conditional Probabilities

Probabilities and expectations under the assumption that some additional temporal condition holds are often needed within the quantitative analysis of protocols. They can provide useful insights concerning the trade-off between different cost and reward functions. For instance, reasoning about the probability of constraints on the energy requirements for reaching a goal, under the condition that the gained utility exceeds a given threshold. The concept of conditional probabilities is also very useful for analyzing a system under the assumption that exceptional events occur. An example is the analysis of fault-tolerant systems, where the impact of failures and the costs for repair mechanisms might be studied under the condition that failures of a certain type occur. For another example, the analysis of a ressource management protocol could be carried out by considering different conditions on the dynamics of the workload.

Conditional Probabilities in Markov Chains. For Markov chains, conditional probabilities can be computed simply by definition as the quotient of ordinary probabilities:

$$\Pr^{\mathcal{M}}_s(\,\varphi \mid \psi\,) \;=\; \frac{\Pr^{\mathcal{M}}_s(\varphi \wedge \psi)}{\Pr^{\mathcal{M}}_s(\psi)}$$

This clearly requires $\Pr^{\mathcal{M}}_s(\psi) > 0$. In what follows, we refer to φ as the *objective* and to ψ as the *condition*. This quotient method has been presented in [4], where the condition and the objective are specified as PCTL path properties. Recently, [28, 33] extended this approach for discrete and continuous-time Markov chains and patterns of path properties with multiple time- and cost-bounds. An alternative approach relies on a transformation $\mathcal{M} \rightsquigarrow \mathcal{M}_\psi$ such that $\Pr^{\mathcal{M}}_s(\,\varphi \mid \psi\,) = \Pr^{\mathcal{M}_\psi}_{s_\psi}(\varphi)$ for all measurable path properties φ [10]. If, e.g., $\psi = \Box B$, then $\mathcal{M}_\psi = (S_\psi, P_\psi)$ with $S_\psi = \{s \in S : \Pr^{\mathcal{M}}_s(\Box B) > 0\}$, $s_\psi = s$, and

$$P_\psi(s,t) \;=\; P(s,t) \cdot \frac{\Pr^{\mathcal{M}}_t(\Box B)}{\Pr^{\mathcal{M}}_s(\Box B)}$$

for all $s, t \in S_\psi$. If $\psi = \Diamond B$, then $\mathcal{M}_\psi$ extends $\mathcal{M}$ by copies $s_\psi = s^{bef}$ of all states s where $\Pr^{\mathcal{M}}_s(\Diamond B) > 0$, with the intuitive meaning "s before B" and the transition probabilities

$$\begin{array}{lcll} P_\psi(s^{bef}, t^{bef}) & = & P(s,t) \cdot \frac{\Pr^{\mathcal{M}}_t(\Diamond B)}{\Pr^{\mathcal{M}}_s(\Diamond B)} & \text{if } s \notin B \\ P_\psi(s^{bef}, t) & = & P(s,t) & \text{if } s \in B \end{array}$$

If ψ is an LTL formula, then we apply standard techniques to construct a deterministic ω-automaton $\mathcal{A}_\psi$ for ψ. Then, the transformation above can be applied for the product Markov chain $\mathcal{M} \otimes \mathcal{A}_\psi$ and the reachability condition $\Diamond B$, where B denotes the union of all bottom strongly connected components of the product in which the acceptance condition of $\mathcal{A}_\psi$ holds. Besides the computation of conditional probabilities for LTL objectives and LTL conditions, this approach is also applicable to compute conditional expected values of random variables in $\mathcal{M}$. For instance, we could compute the (unconditional) expectation of a corresponding random variable in $\mathcal{M}_\psi$. Also conditional quantiles using the techniques sketched in Section 3 such as

$$\begin{aligned} & \min\{\, r \in \mathbb{N} : \Pr_s^{\mathcal{M}}(\,\Diamond^{\leqslant r} B \mid \psi\,) \trianglerighteq q \,\} \\ = \quad & \min\{\, r \in \mathbb{N} : \Pr_{s_\psi}^{\mathcal{M}_\psi}(\Diamond^{\leqslant r} B) \trianglerighteq q \,\} \end{aligned}$$

could be computed easily by the presented approach. Returning to our example, the maximal utility that can be gained with probability 0.8 until reaching a goal state under the condition that only a fixed energy budget e is available is formalized by the quantile:

$$\max\{\, u \in \mathbb{N} : \Pr_{s_{init}}^{\mathcal{M}}(\,\Diamond_{\geqslant u} goal \mid \Diamond^{\leqslant e} goal\,) > 0.8 \}$$

Another example for a performance measure that can provide useful insights for an energy-aware task-scheduling protocol is $\mathrm{E}_{s_{init}}^{\mathcal{M}}[energy \downarrow goal \mid \Diamond_{\geqslant u} goal]$, formalizing the conditional expected energy consumption for reaching the goal, assuming that the achieved utility exceeds u.

Conditional Probabilities in MDPs. The task to reason about extremal conditional probabilities in MDPs is far more challenging. For instance, when $\mathfrak{S}$ ranges over all schedulers with $\Pr_s^{\mathfrak{S}}(\psi) > 0$, one could be interested in computing

$$\Pr_s^{\max}(\,\varphi \mid \psi\,) \;=\; \max_{\mathfrak{S}} \Pr_s^{\mathfrak{S}}(\,\varphi \mid \psi\,) \;=\; \max_{\mathfrak{S}} \frac{\Pr_s^{\mathfrak{S}}(\,\varphi \wedge \psi\,)}{\Pr_s^{\mathfrak{S}}(\,\psi\,)}$$

The crux is that we cannot simply maximize the nominator and denominator independently. This problem has been addressed first in [4], where *conditional PCTL* has been introduced as an extension of PCTL over MDPs by existentially or universally quantified conditional probability constraints $\mathsf{P}_{\bowtie q}(\,\varphi \mid \psi\,)$ for conditional PCTL path formulas φ and ψ. Conditional PCTL has a three-valued semantics, for which we use $\models$, $\not\models$, and $\models\!\!?$ to denote the satisfaction relation, the dissatisfaction relation, and the undefined satisfaction relation, respectively. For instance, if $C = \mathsf{P}_{<q}(\,\varphi \mid \psi\,)$:

- $s \models \exists C$ if $\Pr_s^{\max}(\psi) > 0$ and $\Pr_s^{\min}(\,\varphi \mid \psi\,) < q$
- $s \not\models \exists C$ if $\Pr_s^{\max}(\psi) > 0$ and $\Pr_s^{\min}(\,\varphi \mid \psi\,) \geqslant q$
- $s \models\!\!? \;\exists C$ if $\Pr_s^{\max}(\psi) = 0$ or $t \models\!\!? \;\Phi$ for some state t reachable from s and some proper state subformula Φ of φ or ψ.

For example, the formula $\forall \mathsf{P}_{>0.8}(\Diamond goal | \Diamond failure)$ asserts that even in the worst-case, a goal state will be reached with probability 0.8 under the assumption that a failure will occur. In the context of resilient system analysis [9], the existence of a scheduler with an 80% chance to reach a goal state under the assumption that each failure will eventually be repaired is formalized by the property

$$\exists \mathsf{P}_{>0.8}(\,\Diamond goal \mid \Box(failure \rightarrow \exists \mathsf{P}_{=1}(\Diamond repair))\,)$$

The model-checking algorithm presented in [4], relies on an exhaustive search (with heuristic bounding techniques) in some finite, but potentially exponentially large class of finite-memory schedulers. In [10], we improved this result by presenting a transformation $\mathcal{M} \rightsquigarrow \mathcal{M}_{\varphi|\psi}$ such that

$$\Pr_{\mathcal{M},s_{init}}^{\max}(\,\varphi \mid \psi\,) \;=\; \Pr_{\mathcal{M}_{\varphi|\psi},s_{init}}^{\max}(\varphi)$$

Note that in this approach, we fixed the initial state s_{init} of $\mathcal{M}$. The idea for the construction of the transformed MDP $\mathcal{M}_{\varphi|\psi}$ is to redistribute the probabilities of paths ζ with $\zeta \not\models \psi$ by adding reset transitions to s_{init} from all states s where $\Pr_s^{\min}(\psi) = 0$ (with some fresh action label τ). If $\psi = \Box B$ is an invariance for some $B \subseteq S$, the MDP $\mathcal{M}_{\varphi|\psi}$ results from $\mathcal{M}$ by adding reset transitions from all states $s \in S \backslash B$ to s_{init}. The MDP $\mathcal{M}_{\varphi|\psi}$ then does not rely on the objective φ. The transformation for reachability objectives $\varphi = \Diamond A$ and reachability conditions $\psi = \Diamond B$ is a bit more involved. We first apply some normal-form transformation that permits to assume that all states in A and B are trap states in $\mathcal{M}$. Then, $\mathcal{M}_{\varphi|\psi}$ arises from $\mathcal{M}$ by adding reset transitions from all states s with $\Pr_s^{\min}(\Diamond B) = 0$. To compute maximal conditional probabilities for an LTL objective φ and an LTL condition ψ, we can construct deterministic automata $\mathcal{A}_\varphi$ and $\mathcal{A}_\psi$ for φ and ψ, respectively, and introduce appropriate reset transitions in the product MDP $\mathcal{M} \otimes \mathcal{A}_\varphi \otimes \mathcal{A}_\psi$ to obtain $\mathcal{M}_{\varphi|\psi}$. Within this approach, $\Pr_{\mathcal{M},s_{init}}^{\max}(\,\varphi \mid \psi\,)$ equals the maximal (unconditional) probability for the conjunction of the acceptance conditions of $\mathcal{A}_\varphi$ and $\mathcal{A}_\psi$ in $\mathcal{M}_{\varphi|\psi}$. Minimal conditional probabilities can be computed using the fact that $\Pr_s^{\min}(\,\varphi \mid \psi\,)$ equals $1 - \Pr_s^{\max}(\,\neg\varphi \mid \psi\,)$.

In the worst case, the size of the constructed MDP $\mathcal{M}_{\varphi|\psi}$ grows double exponentially in the lengths of the objective φ and condition ψ and polynomially in the size of $\mathcal{M}$. However, if φ and ψ are (conditional) PCTL formulas and the inner state subformulas of φ and ψ are assumed to be already evaluated such that they can be treated as atomic propositions, the size of $\mathcal{M}_{\varphi|\psi}$ is polynomial in the size of $\mathcal{M}$. Hence, the computational complexity of the model-checking problem for conditional probabilities is the same as in the unconditional case [13, 21]:

Theorem 1 (Complexity) *The model-checking problem for conditional PCTL is in P. The threshold problems for LTL conditions and objectives "does* $\Pr_s^{\min}(\,\varphi \mid \psi\,) \bowtie q$ *hold?" or "does* $\Pr_s^{\max}(\,\varphi \mid \psi\,) \bowtie q$ *hold?" are EXPTIME-complete.*

5. Ratio Objectives

Another important task for the trade-off analysis in probabilistic systems is to establish conditions on the ratio of the accumulated values of two reward functions:

$$ratio \;=\; \frac{rew_1}{rew_2} : FinPaths \rightarrow \mathbb{Q}, \;\; ratio(\pi) = \frac{rew_1(\pi)}{rew_2(\pi)},$$

where $rew_1, rew_2 : S \times Act \rightarrow \mathbb{N}$ are reward functions for a pointed MDP $(\mathcal{M}, s_{init})$ and π ranges over all finite paths with $rew_2(\pi) > 0$. Examples are conditions on the cost-utility quotient, the average recovery time per failure, or the average number of SLA violations per day.

To formalize such ratio conditions within a temporal logical framework, we may extend LTL or PCTL path formulas by atoms of the form $ratio \asymp \vartheta$, where $\vartheta \in \mathbb{Q}$ is a rational threshold and $\asymp$ is one of the comparison operators $\leqslant, <, \geqslant, >$ or $=$. The semantics of these atoms is giveb by interpreting them over path-position pairs:

$$(\zeta, k) \models \; ratio \asymp \vartheta \quad \text{iff} \quad ratio(\zeta[0 \ldots k]) \asymp \vartheta$$

To avoid technical problems with path fragments $\zeta[0 \ldots k]$ where the denominator $rew_2(\zeta[0 \ldots k])$ equals 0, we suppose some default value $\Delta \in \mathbb{Q}$ for the quotient of the accumulated reward for paths of length 0 and suppose that $rew_2(s_{init}, \alpha) > 0$ for all actions $\alpha \in Act(s_{init})$.

We first observe that reasoning about thresholds for ratios of accumulated rewards is closely related to studying assertions $wgt \asymp c$ stating lower or upper bounds for the accumulated value under weight functions. Recall that reward functions are supposed to be non-negative, whereas weight functions can assign negative values to state-action pairs.

Proposition 2 (Ratio vs Weight Constraints) *Decision problems for temporal logics extended by ratio constraints and for temporal logics extended by weight constraints are interreducible.*

Proof. If we are given an MDP with a weight function $wgt : S \times Act \rightarrow \mathbb{Z}$ and an integer $c \in \mathbb{Z}$, constraints of the form $wgt \asymp c$

are reducible to ratio constraints. For example, for the constraint $wgt > 0$ we define two positive reward functions $rew_1, rew_2 : S \times Act \to \mathbb{N}$ by

$$\begin{aligned} rew_1(s,\alpha) &= \max\{+wgt(s,\alpha),0\}+1 \\ rew_2(s,\alpha) &= \max\{-wgt(s,\alpha),0\}+1 \end{aligned}$$

and obtain $wgt = rew_1 - rew_2$. Thus, $wgt(\pi) > 0$ iff $rew_1(\pi) > rew_2(\pi)$ iff $ratio(\pi) > 1$. Hence, weight constraints of the form $wgt > 0$ can be rephrased as ratio constraints $ratio > 1$. The treatment of other weight constraints is analogous.

That a transformation of ratio constraints into weight constraints is possible has been already observed before (see, e.g., [14]). The idea is to replace a given ratio constraint $ratio \asymp \vartheta$ with the constraint $wgt > 0$ for the weight function $wgt : S \times Act \to \mathbb{Z}$ defined by

$$wgt(s,\alpha) \;=\; k \cdot (rew_1(s,\alpha) - \vartheta \cdot rew_2(s,\alpha)),$$

where $k \in \mathbb{Z}$ is the least positive integer such that $k \cdot \vartheta \cdot rew_2(s,\alpha) \in \mathbb{N}$ for all state-action pairs (s,α). Then, $ratio(\pi) \asymp \vartheta$ iff $wgt(\pi) \asymp 0$ for each finite path π of length at least 1. ■

As a consequence of Proposition 2 and the undecidability results for the model-checking problem for temporal logics extended by assertions on the accumulated weights presented in [11, 14], we get the undecidability of the model-checking problem for LTL extended by atoms of the form $ratio \asymp \vartheta$ interpreted over weighted transition systems or weighted Markov chains. Nevertheless, there are decidable algorithmic problems for several ratio constraint patterns.

Probability Constraints on the Ratio. We first consider invariances stating that the ratio of accumulated rewards always exceeds a given rational threshold ϑ in combination with an ω-regular path property. More precisely, we address the task to check whether $\Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}(\Box(ratio > \vartheta) \wedge \varphi) \trianglerighteq q$ for some/all schedulers, where q is a rational probability bound and φ is an LTL formula.

For instance, let us consider a server system where for a scheduler $\mathfrak{S}$ we are interested in the probability that an availability of 99% is guaranteed and any failure is fixed within at most t seconds. This probability can be expressed following the ratio objective pattern stated above:

$$\Pr^{\mathfrak{S}}_{\mathcal{M},s_{init}}\big(\Box\,(\mathit{rel\_avail} > 0.99) \wedge \Box(\mathit{failure} \Longrightarrow \Diamond^{\leq t}\mathit{repair})\big),$$

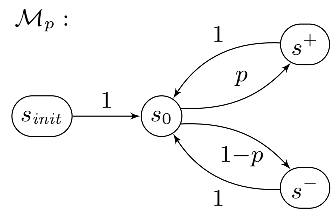
where the relative availability time of the server is the ratio

$$\mathit{rel\_avail} \;=\; \frac{\mathit{total\_time} - \mathit{failure\_time}}{\mathit{total\_time}}$$

which employs the reward functions *failure_time* and *total_time*. A similar pattern turned out to be important in the context of resilient system analysis [9].

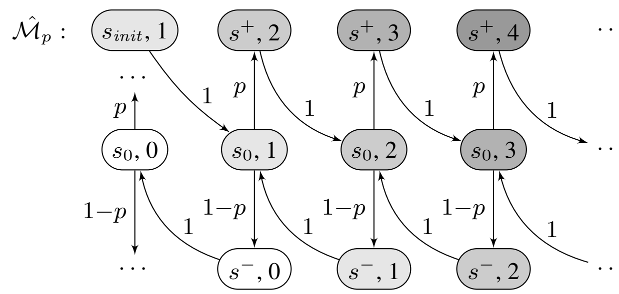
The transformation of ratio constraints into weight constraints of the form $wgt > 0$ sketched in the proof of Proposition 2 can be used to reduce several algorithmic problems for ratio constraints into corresponding termination problems for pushdown automata over a single stack symbol (also called *one-counter machines*). The simple idea is to mimic each α-transition from a state s by k push resp. pop operations, provided that $k = |wgt(s,\alpha)| > 0$. Then, the weight constraint $wgt > 0$ is equivalent to the requirement that the stack is nonempty. This observation yields the decidability of several algorithmic problems for ratio constraints [8]. For example, if $\mathcal{M}$ is a Markov chain, then the task to check whether $\Pr^{\mathcal{M}}_{s_{init}}(\Box(ratio > \vartheta) \wedge \varphi) \bowtie q$ is solvable using known algorithms for probabilistic pushdown systems [15] if φ is an LTL formula and q is a rational probability bound. Although this reduction causes an exponential blow-up when the size of the weights is determined by the number of digits in their representation as decimal or binary numbers, we cannot expect efficient algorithms. This is due to the fact that even the problem to check whether $\Box(ratio \asymp \vartheta)$ holds with positive probability depends on the precise transition probabilities in $\mathcal{M}$, whereas, e.g., the problems to check whether $\Pr^{\mathcal{M}}_s(\varphi) > 0$ for some LTL formula φ or whether $\Pr^{\mathcal{M}}_s(A\mathrm{U}^{\leqslant r}B) > 0$ only depend on the graph structure of $\mathcal{M}$, but not on the concrete transition probabilities.

Consider for instance the Markov chain $\mathcal{M} = \mathcal{M}_p$ depicted in the following picture, where $p \in \,]0,1[$ is a probability parameter. Since action labels are irrelevant in Markov chains, we attach weights to the states that, following the approach of Proposition 2, are obtained from the constraint $ratio > 1$ when $rew_1(s) = 2$, $rew_2(s) = 1$ for $s \in \{s_{init}, s^+\}$, $rew_1(s_0) = rew_2(s_0) = 1$ and $rew_1(s^-) = 1$, $rew_2(s^-) = 2$.



$$\begin{aligned} wgt(s_{init}) &= +1 \\ wgt(s_0) &= \;\;0 \\ wgt(s^+) &= +1 \\ wgt(s^-) &= -1 \end{aligned}$$

By unfolding $\mathcal{M}$ from the initial state s_{init} into an infinite Markov chain over the finite paths in $\mathcal{M}$, we can obtain a Markov chain $\hat{\mathcal{M}}$ constituting a biased random walk: The states of $\hat{\mathcal{M}}$ are pairs (s,c), where s is a state of $\mathcal{M}$ and $c \in \mathbb{Z}$ is a weight, representing all finite paths in the unfolding of $\mathcal{M}$ which end in s and have the accumulated weight c.



For this biased random walk it is well-known that with $p > \frac{1}{2}$, it drifts to the right and never reaches a state having a negative accumulated weight assigned with positive probability, whereas for $p \leqslant \frac{1}{2}$, the states with negative accumulated weight will be visited almost surely. Thus, $\Pr^{\mathcal{M}}_{s_0}(\Box(ratio > 1)) > 0$ iff $\Pr^{\mathcal{M}}_{s_0}(\Box(wgt > 0)) > 0$ iff $p > \frac{1}{2}$.

More efficient algorithms can be designed for almost-sure constraints. If $\mathcal{M}$ is an MDP, then the problem to find a scheduler $\mathfrak{S}$ with $\Pr^{\mathfrak{S}}_s(\Box(ratio > \vartheta) \wedge \varphi) = 1$ is in NP ∩ coNP and solvable via known algorithms for energy Büchi games [16, 17]. As shown in [8], the universal problem that asks whether $\Pr^{\mathfrak{S}}_s(\Box(ratio \asymp \vartheta) \wedge \varphi) = 1$ for all schedulers $\mathfrak{S}$ is even simpler and solvable in P as we have

$$\Pr^{\min}_s(\Box(ratio \asymp \vartheta) \wedge \varphi) = 1 \quad \text{iff} \quad s \not\models \exists\Diamond(wgt \leqslant 0),$$

where wgt is as in the proof of Theorem 2. The latter existential formula can be checked efficiently by standard shortest-path algorithms (e.g., the Bellman-Ford algorithm). Similarly, the almost-sure ratio problem for some fixed window size $\ell \in \mathbb{N}$ asking whether

$$\Pr^{\max}_s\big\{\, \zeta \,:\, ratio(\zeta[i \ldots i{+}\ell]) \asymp \vartheta \text{ for all } i \in \mathbb{N} \,\big\} \;=\; 1$$

is solvable in polynomial time by shortest-path algorithms [11].

Expected Long-run Ratio. The expected long-run ratio of two reward functions rew_1 and rew_2 is defined as the expectation of the random variable $\mathbb{L}[ratio] : InfPaths(s_{init}) \to \mathbb{R}$ given by:

$$\mathbb{L}[ratio](\zeta) \;=\; \limsup_{n\to\infty} \frac{rew_1(\zeta[0\ldots n])}{rew_2(\zeta[0\ldots n])}$$

Let us first suppose that $\mathcal{M}$ is a Markov chain, in which case rew_1 and rew_2 can be viewed as functions $rew_i : S \to \mathbb{N}$. It is well-known that almost all infinite paths eventually enter a bottom strongly connected component (BSCC) C and visit all states of C infinitely often. More precisely, for almost all infinite paths ζ with $\zeta \models \Diamond C$, the frequency of visiting state $s \in C$ is the steady-state probability $\mathbb{S}_C(s)$ of state s inside C is

$$\lim_{n\to\infty} \frac{1}{n+1} \big| \{ k \in \{0,1,\ldots,n\} : \zeta[k] = s \} \big| \;=\; \mathbb{S}_C(s),$$

where the vector $\big(\mathbb{S}_C(s)\big)_{s\in C}$ is the unique solution of the flow equations $\sum_{s'\in C} P(s',s) \cdot \mathbb{S}_C(s') = \mathbb{S}_C(s)$ with the side-constraint $\sum_{s\in C} \mathbb{S}_C(s) = 1$. Then, almost all paths ζ with $\zeta \models \Diamond C$ have the same long-run ratio, namely:

$$\mathbb{L}_C[ratio] = \frac{\mathbb{L}_C[rew_1]}{\mathbb{L}_C[rew_2]}, \text{ where } \mathbb{L}_C[rew_i] = \sum_{s\in C} \mathbb{S}_C(s) \cdot rew_i(s),$$

provided that $rew_2(s) > 0$ for at least one state $s \in C$. Then, the expected long-run ratio in the pointed Markov chain $(\mathcal{M}, s_{init})$ is:

$$\mathrm{E}^{\mathcal{M}}_{s_{init}}(\mathbb{L}[ratio]) \;=\; \sum_{C} \Pr^{\mathcal{M}}_{s_{init}}(\Diamond C) \cdot \mathbb{L}_C[ratio],$$

where C ranges over all BSCCs of $\mathcal{M}$.

Theorem 3 *The expected long-run ratio of a Markov chain $\mathcal{M}$ can be computed in time polynomial in the size of $\mathcal{M}$.*

Obviously, if $\mathcal{M}$ is strongly connected, then $\mathcal{M}$ consists of a single BSCC, in which case the long-run expected ratio does not depend on the initial state s_{init}. In this case, we simply write $\mathrm{E}^{\mathcal{M}}(\mathbb{L}[ratio])$. Extremal expected long-run ratios in MDPs have been addressed in [2, 42]. In what follows, we suppose that each end component contains at least one state-action pair (s,α) with $rew_2(s,\alpha) > 0$. This assumption ensures the existence of the expectation of the random variable $\mathbb{L}[ratio]$ under each scheduler. Using the results of [29], [42] proves the existence of a memoryless deterministic scheduler that minimizes the expected long-run ratio. Furthermore, [42] presents a characterization of the minimal expected long-run ratio as a linear fractional program as well as an iterative linear-programming approximation scheme for unichain MDPs, i.e., MDPs such that the induced (finite-state) Markov chain of each memoryless scheduler is strongly connected. We suggest here a variant of this approach to compute the maximal and minimal expected long-run ratio for an arbitrary (possibly non-unichain) MDP. Obviously, if $\mathfrak{E} = (T, \mathfrak{A})$ is an end component of $\mathcal{M}$, where $\mathfrak{A}(t) = \{\alpha_t\}$ is a singleton for all states $t \in T$, then $\mathcal{M}_{\mathfrak{E}} = (T, P_{\mathfrak{E}})$ with $P_{\mathfrak{E}}(t,t') = P(t,\alpha_t,t')$ is a strongly connected Markov chain. For Markov chains, we can apply the method sketched above and compute the expected long-run ratio $r_i = \mathrm{E}^{\mathcal{M}_{\mathfrak{E}_i}}(\mathbb{L}[ratio])$ for $i = 1,\ldots,k$. Let now $\mathfrak{E}_1,\ldots,\mathfrak{E}_k$ be an enumeration of the end components $\mathfrak{E} = (T,\mathfrak{A})$, where $|\mathfrak{A}(t)| = 1$ for all states $t \in T$. We define $\mathcal{M}'$ to be the MDP that results from $\mathcal{M}$ by adding fresh action symbols $\tau, \tau_1, \ldots, \tau_k$ and a fresh trap state $goal$ with $P'(s,\tau_i,goal) = 1$ if s belongs to $\mathfrak{E}_i$ and $P'(s,\tau_i,\cdot) = 0$ in all other cases. The new MDP $\mathcal{M}'$ is equipped with the reward function rew' given by $rew'(s,\tau_i) = r_i$ if s belongs to $\mathfrak{E}_i$ and $rew'(\cdot) = 0$ in all other cases. Using the result of [42] stating the existence of optimal deterministic memoryless schedulers for expected long-run ratio objectives, we obtain

$$\mathrm{E}^{\max}_{\mathcal{M},s_{init}}(\mathbb{L}[ratio]) \;=\; \mathrm{E}^{\max}_{\mathcal{M}',s_{init}}[rew' \!\downarrow\! goal]$$

and the analogous statement for the minimal expected long-run ratio. Although the latter can be computed by standard linear-programming techniques in time polynomial in the size of $\mathcal{M}'$, the suggested approach is computationally expensive as the number of end component can grow exponentially. For the *expected long-run threshold problem*

“is there a scheduler $\mathfrak{S}$ with $\mathrm{E}^{\mathfrak{S}}_{s_{init}}(\mathbb{L}[ratio]) \asymp \vartheta$?”,

where as before ϑ is some rational threshold and $\asymp \in \{\leqslant, <, \geqslant, >, =\}$, we can guess nondeterministically a deterministic memoryless scheduler $\mathfrak{S}$ and check whether its expected long-run ratio meets the bound $\asymp \vartheta$. This yields:

Theorem 4 *The expected long-run threshold problem is in NP.*

To the best of our knowledge, the precise complexity of the expected long-run problem is an open question left for further work.

6. Conclusions

In this article, we reported on our current work on algorithmic problems for discrete Markovian models that appeared to us when applying probabilistic model-checking in inter-disciplinary projects. We addressed quantiles, conditional probabilities and ratio constraints for accumulated cost or reward functions for analyzing the interplay between multiple objectives. For quantiles and conditional probabilities, we already carried out prototype implementations [7, 10] based on PRISM, which we could use in case studies from different domains (e.g., [9, 25]). We are currently working on a prototype implementation in the context of ratios, involving several heuristics. Regarding the theory, there are still various interesting problems left open, including complexity-theoretic considerations and extensions for continuous-time models and other probabilistic real-time models such as probabilistic timed automata. Given the Turing power of two-counter machines, undecidability results for algorithmic problems for structures with two weight functions are no surprise. However, we found it remarkable that the switch from a single (non-negative) reward functions to integer weight function has a drastic effect. Even apparently simple problems, such as the task to compute the probability for a weight invariance $\Box(wgt > 0)$ in a Markov chain, turn out to be hard. Our work on ratios in Markovian models is in the line of a current research trend to extend temporal logics, transition systems and game structures with weight functions, see, e.g., [1, 11, 12, 14, 16, 17, 34, 39].

References

[1] P. A. Abdulla, R. Mayr, A. Sangnier, and J. Sproston. Solving parity games on integer vectors. In *24th International Conference on Concurrency Theory (CONCUR)*, volume 8052 of *Lecture Notes in Computer Science*, pages 106–120. Springer, 2013.

[2] V. Aggarwal, R. Chandrasekaran, and K. Nair. Markov ratio decision processes. *Journal of Optimization Theory and Application*, 21(1), 1977.

[3] S. Andova, H. Hermanns, and J.-P. Katoen. Discrete-time rewards model-checked. In *First International Workshop on Formal Modeling and Analysis of Timed Systems Workshop (FORMATS)*, volume 2791 of *Lecture Notes in Computer Science*, pages 88–104. Springer, 2003.

[4] M. Andrés and P. van Rossum. Conditional probabilities over probabilistic and nondeterministic systems. In *14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 4963 of *Lecture Notes in Computer Science*, pages 157–172. Springer, 2008.

[5] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[6] C. Baier and M. Kwiatkoswka. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11(3):125–155, 1998.

[7] C. Baier, M. Daum, C. Dubslaff, J. Klein, and S. Klüppelholz. Energy-utility quantiles. In *6th NASA Formal Methods Symposium (NFM)*, volume 8430 of *Lecture Notes in Computer Science*, pages 285–299. Springer, 2014.

[8] C. Baier, C. Dubslaff, J. Klein, S. Klüppelholz, and S. Wunderlich. Probabilistic model checking for energy-utility analysis. In *Festschrift for Prakash Panagaden's 60th Birthday*, volume 8464 of *Lecture Notes in Computer Science*. Springer, 2014.

[9] C. Baier, C. Dubslaff, S. Klüppelholz, and L. Leuschner. Energy-Utility Analysis for Resilient Systems Using Probabilistic Model Checking. In *Proceedings of the 35th International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS'14)*, volume 8489 of *Lecture Notes in Computer Science*, pages 20–39. Springer, 2014.

[10] C. Baier, J. Klein, S. Klüppelholz, and S. Märcker. Computing conditional probabilities in Markovian models efficiently. In *20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 8413 of *Lecture Notes in Computer Science*, pages 515–530. Springer, 2014.

[11] C. Baier, J. Klein, S. Klüppelholz, and S. Wunderlich. Weight monitoring with linear temporal logic: Complexity and decidability. In *23rd EACSL Annual Conference on Computer Science Logic and 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS-CSL)*. ACM, 2014.

[12] S. S. Bauer, L. Juhl, K. G. Larsen, J. Srba, and A. Legay. A logic for accumulated-weight reasoning on multiweighted modal automata. In *Sixth International Symposium on Theoretical Aspects of Software Engineering (TASE)*, pages 77–84. IEEE, 2012.

[13] A. Bianco and L. de Alfaro. Model checking of probabilistic and non-deterministic systems. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 1026 of *Lecture Notes in Computer Science*, pages 499–513. Springer, 1995.

[14] U. Boker, K. Chatterjee, T. A. Henzinger, and O. Kupferman. Temporal specifications with accumulative values. In *26th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 43–52. IEEE Computer Society, 2011.

[15] T. Brázdil, A. Kučera, and O. Stražovský. On the decidability of temporal properties of probabilistic pushdown automata. In *22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 3404 of *Lecture Notes in Computer Science*, pages 145–157. Springer, 2005.

[16] K. Chatterjee and L. Doyen. Energy and mean-payoff parity Markov decision processes. In *36th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 6907 of *Lecture Notes in Computer Science*, pages 206–218. Springer, 2011.

[17] K. Chatterjee and L. Doyen. Energy parity games. *Theoretical Computer Science*, 458:49–60, 2012.

[18] K. Chatterjee and M. Henzinger. Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification. In *22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1318–1336. SIAM, 2011.

[19] K. Chatterjee, R. Majumdar, and T. Henzinger. Markov decision processes with multiple objectives. In *23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 3884 of *Lecture Notes in Computer Science*, pages 325–336. Springer, 2006.

[20] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 2000.

[21] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[22] L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, Department of Computer Science, 1997.

[23] L. de Alfaro. How to specify and verify the long-run average behavior of probabilistic systems. In *13th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 454–465. IEEE Computer Society, 1998.

[24] L. de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In *10th International Conference on Concurrency Theory (CONCUR)*, volume 1664 of *Lecture Notes in Computer Science*, pages 66–81. Springer, 1999.

[25] C. Dubslaff, S. Klüppelholz, and C. Baier. Probabilistic Model Checking for Energy Analysis in Software Product Lines. In *Proceedings of the 13th International Conference on Modularity*, MODULARITY '14, pages 169–180. ACM, 2014.

[26] K. Etessami, M. Kwiatkowska, M. Vardi, and M. Yannakakis. Multi-objective model checking of Markov decision processes. *Logical Methods in Computer Science*, 4(4), 2008.

[27] V. Forejt, M. Kwiatkowska, G. Norman, and D. Parker. Automated verification techniques for probabilistic systems. In *11th International School on Formal Methods for the Design of Computer, Communication and Software Systems (SFM)*, volume 6659 of *Lecture Notes in Computer Science*, pages 53–113. Springer, 2011.

[28] Y. Gao, M. Xu, N. Zhan, and L. Zhang. Model checking conditional CSL for continuous-time Markov chains. *Information Processing Letters*, 113(1–2):44 – 50, 2013.

[29] H. Gimbert. Pure stationary optimal strategies in Markov decision processes. In *24th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 4393 of *Lecture Notes in Computer Science*, pages 200–211. Springer, 2007.

[30] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*, 2002. Springer.

[31] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.

[32] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A tool for automatic verification of probabilistic systems. In *12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 3920 of *Lecture Notes in Computer Science*, pages 441–444. Springer, 2006.

[33] M. Ji, D. Wu, and Z. Chen. Verification method of conditional probability based on automaton. *Journal of Networks*, 8(6):1329–1335, 2013.

[34] L. Juhl, K. G. Larsen, and J.-F. Raskin. Optimal bounds for multiweighted and parametrised energy games. In *Theories of Programming and Formal Methods - Essays Dedicated to Jifeng He on the Occasion of His 70th Birthday*, volume 8051 of *Lecture Notes in Computer Science*, pages 244–255. Springer, 2013.

[35] J.-P. Katoen, I. Zapreev, E. Hahn, H. Hermanns, and D. Jansen. The ins and outs of the probabilistic model checker MRMC. *Performance Evaluation*, 68(2), 2011.

[36] V. Kulkarni. *Modeling and Analysis of Stochastic Systems*. Chapman & Hall, 1995.

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

[38] R. J. Serfling. *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons, 1980.

[39] T. Tomita, S. Hiura, S. Hagihara, and N. Yonezaki. A temporal logic with mean-payoff constraints. In *14th International Conference on Formal Engineering Methods. Formal Methods and Software Engineering (ICFEM)*, volume 7635 of *Lecture Notes in Computer Science*, pages 249–265. Springer, 2012.

[40] M. Ummels and C. Baier. Computing quantiles in Markov reward models. In *16th International Conference on Foundations of Software Science and Computation Structures (FOSSACS)*, volume 7794 of *Lecture Notes in Computer Science*, pages 353–368. Springer, 2013.

[41] M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *26th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 327–338. IEEE Computer Society, 1985.

[42] C. von Essen and B. Jobstmann. Synthesizing systems with optimal average-case behavior for ratio objectives. In *International Workshop on Interactions, Games and Protocols (iWIGP)*, volume 50 of *EPTCS*, pages 17–32, 2011.