Project no.:	**PIRSES-GA-2011-295261**
Project full title:	**Mobility between Europe and Argentina applying Logics to Systems**
Project Acronym:	**MEALS**
Deliverable no.:	**1.4 / 1**
Title of Deliverable:	**Energy-Utility Quantiles**

Contractual Date of Delivery to the CEC:	**31-Mar-2014**
Actual Date of Delivery to the CEC:	**31-May-2014**
Organisation name of lead contractor for this deliverable:	**UNC**
Author(s):	**Christel Baier, Marcus Daum, Clemens Dubslaff, Joachim Klein, Sascha Klüppelholz**
Participants(s):	**TUD**
Work package contributing to the deliverable:	**WP1**
Nature:	**R**
Dissemination Level:	**Public**
Total number of pages:	**17**
Start date of project:	1 Oct. 2011 **Duration:** 48 month

Abstract:

The concept of quantiles is well-known in statistics, but its benefits for the formal quantitative analysis of probabilistic systems have been noticed only recently. To compute quantiles in Markov decision processes where the objective is a probability constraint for an until (i.e., constrained reachability) property with an upper reward bound, an iterative linear-programming (LP) approach has been proposed in a recent paper. We consider here a more general class of quantiles with probability or expectation objectives, allowing to reason about the trade-off between costs in terms of energy and some utility measure. We show how the iterative LP approach can be adapted for these types of quantiles and propose another iterative approach that decomposes the LP to be solved into smaller ones. This algorithm has been implemented and evaluated in case studies for quantiles where the objective is a probability constraint for until properties with upper reward bounds.

Note:

This deliverable is based on material that has been published in J.M. Badger and K.Y. Rozier (Eds.): NFM 2014, LNCS 8430, pp. 285–299, Springer, 2014. Available at `http://dx.doi.org/10.1007/978-3-319-06200-6_24`.

This project has received funding from the European Union Seventh Framework Programme (FP7 2007-2013) under Grant Agreement Nr. 295261.

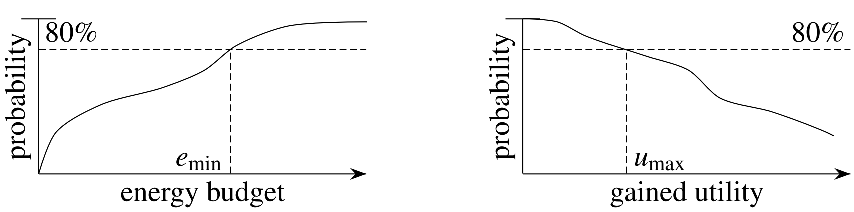
Contents

1 Introduction

The concept of quantiles is well-known in statistics (see, e.g., [21]) and used there to reason about the cumulative distribution function of a random variable R. Quantiles are defined as maximal values r such that the probability for the event $R > r$ is beyond a given threshold. Although quantiles can provide very useful insights in the interplay of various cost functions and other system properties, they have barely obtained attention in the context of formal algorithmic system analysis. Quantiles for probabilistic operational models, such as Markov chains or Markov decision processes, can be defined using parameterized state properties $\Phi[r]$ or $\Psi[r]$, where r is a parameter for some cost or reward function and $\Phi[r]$ is increasing in r, whereas $\Psi[r]$ is decreasing in r. The notion "increasing" means that $s \models \Phi[r]$ implies $s \models \Phi[i]$ for all $i > r$ ("decreasing" has an analogous meaning). Quantiles for objectives $\Phi[r]$ and $\Psi[r]$ in state s of the given model are defined as $\min\{\, r \;:\; s \models \Phi[r]\,\}$ resp. $\max\{\, r \;:\; s \models \Psi[r]\,\}$. We formalize $\Phi[r]$ and $\Psi[r]$ by PRCTL-like constraints that assert lower or upper bounds either for the probabilities for reward-bounded path formulas or for the expected accumulated rewards until reaching a certain target. Typical examples are formulas of the form $\Phi_u[e]$ for fixed u and $\Psi_e[u]$ for fixed e asserting that the probability for

$$\lambda_{e,u} \;=\; \Diamond(\,(\mathit{energy} \leqslant e) \wedge (\mathit{utility} \geqslant u)\,)$$

is, e.g., at least 0.8. (We use LTL notations where the temporal operator $\Diamond$ stands for "eventually".) The quantile $e_{\min} = \min\{e \in \mathbb{N} : s \models \Phi_u[e]\}$ is the minimal initial energy budget required to achieve the utility value u with probability at least 0.8, while $u_{\max} = \max\{u \in \mathbb{N} : s \models \Psi_e[u]\}$ is the maximal utility that can be achieved with probability at least 0.8, when the energy budget is e. The curve for $\lambda_{e,u}$ on the left of the figure below illustrates how the probability increases when the utility value u is fixed and the energy budget e tends to ∞. The curve on the right shows how the probability for $\lambda_{e,u}$ decreases when the energy budget e is fixed and the demanded degree of utility tends to ∞.



State properties $\Phi[r]$ or $\Psi[r]$ can also impose a constraint on the expected value of a random variable. For example, one might ask for the minimal initial energy budget e that is needed to ensure that the expected degree of utility is at least some predefined utility threshold u. Vice versa, an expectation quantile might specify the maximal degree of utility that can be achieved when the expected energy consumption is required to be less or equal some fixed value e.

In probabilistic models with nondeterminism (e.g., for modeling concurrency by interleaving) such as Markov decision processes (MDPs), quantiles can be defined either in an existential or in a universal version, depending on whether the quantile is used in a worst-case analysis (where all possible resolutions of the nondeterminism are taken into account) or whether the task is to synthesize a control mechanism that schedules actions in an optimal way.

As the above examples suggest, quantiles can be seen as a concept to reason about the trade-off between different quantitative aspects, such as energy and utility. Thus, they yield an alternative to *multi-objective reasoning* for MDPs by means of Pareto optimal schedulers for multiple objectives given as Boolean combinations of constraints on the probabilities for certain events and/or expected accumulated costs [11, 12]. The demand for algorithms to compute quantiles in Markovian models occurred to us during case studies with resource management protocols [3]. However, in various case studies with probabilistic model checkers carried out by other researchers, quantiles have been used implicitly in diagrams illustrating the evaluation results of the experimental studies.

Model-checking algorithms for various types of properties with *fixed* reward bounds have been proposed for discrete Markovian models and implemented in tools, see, e.g., [1, 18, 15]. The task to compute quantiles is, however, more challenging since it requires to compute an *optimal* reward bound for parameterized objectives. Our recent paper [4] briefly considers quantiles for discrete and continuous-time Markov chains, as an example for nonstandard multi-objective reasoning. To the best of our knowledge, [22] is the only paper where the computation of quantiles has been addressed for MDPs. It considers quantiles in MDPs with a nonnegative reward function for the states where the objective is a probability constraint for a reachability property with an upper reward bound r, formalized using the temporal reward-bounded until operator $\mathrm{U}^{\leqslant r}$. The above mentioned quantile $\min\{e : s \models \Phi_u[e]\}$ appears as a special case since $\Phi_u[e]$ can be seen as a probability constraint for the path property $\lambda_{e,u} = \Diamond^{\leqslant e}(\mathit{utility} \geqslant u) = \mathit{true}\,\mathrm{U}^{\leqslant e}(\mathit{utility} \geqslant u)$. In [22], polynomial-time algorithms for qualitative constraints where the probability bounds are 0 or 1 and an iterative linear-programming (LP) approach for probability bounds p with $0 < p < 1$ has been presented. The minimal or maximal probabilities for a path property $A\,\mathrm{U}^{\leqslant r}\,B$ for $r = 0, 1, 2, \ldots$ is calculated until the probability bound p is reached, where the extrema are taken over all resolutions of nondeterminism. This approach appears to be naïve, but the computation of quantiles is known to be computationally hard (at least NP-hard already for Markov chains by the results of [19]). This is reflected in the exponential upper bound in [22] for the number of iterations and the size of the LPs to be solved.

Contribution. First, we generalize the approach of [22] by introducing general notions of quantiles in MDPs where the objective can either be a probability constraint or a constraint on an expectation (Sec. 3). Second, we revisit the iterative LP approach suggested by [22] and discuss refinements that make the approach feasible in practice. The core idea is an iterative method that propagates intermediate results as much as possible and follows the dynamic-programming scheme with embedded LPs to deal with zero-reward cycles (Sec. 4.2). We implemented this approach into Prism [14] and study its performance by means of an energy-aware job-scheduling system (Sec. 6). Third, we present new algorithms for the computation of quantiles in MDPs where the objective is (a) either a probability constraint for reachability conditions with lower reward bounds (Sec. 4.3), or (b) a constraint on the expected accumulated reward (Sec. 5). These algorithms also rely on an iterative LP approach and the propagation principle is applicable as well (Sec. 4). Although we are not aware that expectation quantiles in MDPs have been addressed before, the presented algorithm for (b) shares some similarities with algorithms that have been proposed for stochastic shortest path problems [6] and to maximize/minimize the expected cost to reach a target [10].

2 Preliminaries

We provide a brief summary of the relevant concepts of MDPs and specifications given as formulas in probabilistic computation tree logic with reward-bounded modalities (PRCTL). Further details can be found, e.g., in [20, 9, 5].

Markov decision processes (MDPs). An MDP is a tuple $\mathcal{M} = (S, Act, P)$, where S is a finite set of states, Act a finite set of actions, $P : S \times Act \times S \to [0,1]$ such that $\sum_{s' \in S} P(s, \alpha, s') \in \{0,1\}$ for all states $s \in S$ and actions $\alpha \in Act$. The tuples $(s, \alpha, s') \in S \times Act \times S$ with $P(s, \alpha, s') > 0$ are called *steps* and we then say that state s' is an α-successor of s. We write $Act(s)$ for the set of actions α that have an α-successor from state $s \in S$ and require that $Act(s) \neq \varnothing$ for all states s. Intuitively, if the current state of $\mathcal{M}$ is s, then first there is a nondeterministic choice to select one of the enabled actions α. Then, $\mathcal{M}$ behaves probabilistically and moves with probability $P(s, \alpha, s')$ to some state s'. *Markov chains* are purely probabilistic instances of MDPs, i.e., where the action set is a singleton.

Paths in an MDP $\mathcal{M}$ can be seen as sample runs with resolved nondeterminism. Formally, paths are finite or infinite sequences $\pi = s_0 \, \alpha_0 \, s_1 \, \alpha_1 \, s_2 \, \alpha_2 \ldots \in (S \times Act)^* S \cup (S \times Act)^\omega$ that are built by consecutive steps, i.e., $\alpha_i \in Act(s_i)$ and $P(s_i, \alpha_i, s_{i+1}) > 0$ for all i. $\pi[k]$ denotes the $(k{+}1)$-st state in π and $pref(\pi, k)$ the prefix of π consisting of the first k steps, ending in state $\pi[k] = s_k$. We write $FPaths(s)$ for the set of finite paths and $IPaths(s)$ for the set of infinite paths starting in s.

Reward structure. A reward structure $\mathcal{R}$ for $\mathcal{M}$ consists of finitely many reward functions $rew : S \times Act \to \mathbb{N}$. If $\pi = s_0 \, \alpha_0 \, s_1 \, \alpha_1 \ldots \alpha_{n-1} \, s_n$ is a finite path, then the accumulated reward $rew(\pi)$ is the sum of the rewards for the state-action pairs, i.e., $rew(\pi) = \sum_{0 \leqslant i < n} rew(s_i, \alpha_i)$.

Schedulers and induced probability space. Reasoning about probabilities for path properties in MDPs requires the selection of an initial state and the resolution of the nondeterministic choices between the possible transitions. The latter is formalized via *schedulers*, often also called policies or adversaries, which take as input a finite path and select an action to be executed. A (deterministic) scheduler is a function $\mathfrak{S} : FPaths \to Act$ such that $\mathfrak{S}(\pi) \in Act(s_n)$ for all finite paths $\pi = s_0 \, \alpha_0 \ldots \alpha_{n-1} \, s_n$. An $\mathfrak{S}$-*path* is any path that arises when the nondeterministic choices in $\mathcal{M}$ are resolved using $\mathfrak{S}$, i.e., $\mathfrak{S}(pref(\pi, k)) = \alpha_k$ for all $0 \leqslant k < n$. Infinite $\mathfrak{S}$-paths are defined accordingly. Given some scheduler $\mathfrak{S}$ and state s (viewed as the initial state), the behavior of $\mathcal{M}$ under $\mathfrak{S}$ is purely probabilistic and can be formalized by a tree-like (infinite-state) Markov chain $\mathcal{M}_s^{\mathfrak{S}}$. One can think of the states in $\mathcal{M}_s^{\mathfrak{S}}$ as finite $\mathfrak{S}$-paths $\pi = s_0 \alpha_0 \ldots \alpha_{n-1} s_n$ starting in state s, where the probability to move from π to $\pi \, \alpha \, s'$ is simply $P(s_n, \alpha, s')$. Using standard concepts of measure and probability theory, a sigma-algebra and a probability measure $\mathrm{Pr}_s^{\mathfrak{S}}$ for measurable sets of the infinite paths in the Markov chain $\mathcal{M}_s^{\mathfrak{S}}$, also called *(path) events* or *path properties*, is defined and can be transferred to maximal $\mathfrak{S}$-paths in $\mathcal{M}$ starting in s. For further details, we refer to standard text books such as [13, 16, 20].

For a worst-case analysis of a system modeled by an MDP $\mathcal{M}$, one ranges over all initial states and all schedulers (i.e., all possible resolutions of the nondeterminism) and considers the minimal or maximal probabilities for φ. If φ represents a desired path property, then $\mathsf{Pr}_s^{\min}(\varphi) = \inf_{\mathfrak{S}} \mathsf{Pr}_s^{\mathfrak{S}}(\varphi)$ is the probability for $\mathcal{M}$ satisfying φ that can be guaranteed even for worst-case scenarios, i.e., when ranging over all schedulers. Similarly, if φ stands for a bad (undesired) path property, then $\mathsf{Pr}_s^{\max}(\varphi) = \sup_{\mathfrak{S}} \mathsf{Pr}_s^{\mathfrak{S}}(\varphi)$ is the least upper bound that can be guaranteed for the bad behaviors.

State and path properties. Let s be a state, $p \in [0,1]$ a probability bound, $\bowtie \in \{<, \leqslant, \geqslant, >\}$ and φ a path property. We write $s \models \exists \mathrm{P}_{\bowtie p}(\varphi)$ if there exists a scheduler $\mathfrak{S}$ with $\mathsf{Pr}_s^{\mathfrak{S}}(\varphi) \bowtie p$. Similarly, $s \models \forall \mathrm{P}_{\bowtie p}(\varphi)$ if $\mathsf{Pr}_s^{\mathfrak{S}}(\varphi) \bowtie p$ for all schedulers $\mathfrak{S}$. Given a reward structure $\mathcal{R}$ with reward function *rew*, sets A, $B \subseteq S$, and $r \in \mathbb{N}$, then $A \,\mathrm{U}(\, (\mathit{rew} \bowtie r) \wedge B\,)$ stands for the set of infinite paths $\tilde{\pi}$ such that there is some $k \in \mathbb{N}$ with $\mathit{rew}(\,\mathit{pref}(\tilde{\pi}, k)\,) \bowtie r$ and $\tilde{\pi}[k] \in B$, $\tilde{\pi}[i] \in A$ for $0 \leqslant i < k$. If *rew* is clear from the context (e.g., if the reward structure $\mathcal{R}$ is a singleton), we briefly write $A \,\mathrm{U}^{\bowtie r} B$ rather than $A \,\mathrm{U}((\mathit{rew} \bowtie r) \wedge B)$. We often use the notation $\pi \models A \,\mathrm{U}^{\bowtie r} B$ instead of $\pi \in A \,\mathrm{U}^{\bowtie r} B$. As usual, we derive the release operator R by $A \,\mathrm{R}^{\bowtie r} B = \neg(\neg A \,\mathrm{U}^{\bowtie r} \neg B)$, where $\neg B$ denotes the complement of B. The temporal modalities $\Diamond$ (eventually) and $\Box$ (always) with or without reward-bounds are derived as usual, e.g., $\Diamond^{\bowtie r} B = \mathit{true} \,\mathrm{U}^{\bowtie r} B$ and $\Box^{\bowtie r} B = \neg \Diamond^{\bowtie r} \neg B$, where *true* stands for the full state space.

Reward-bounded path properties such as $\varphi[r] = A \,\mathrm{U}^{\leqslant r} B$ are called *increasing* as $\tilde{\pi} \models \varphi[r]$ implies $\tilde{\pi} \models \varphi[r{+}1]$. The dual path properties $\psi[r] = \neg\varphi[r]$ are called *decreasing* as $\tilde{\pi} \models \psi[r{+}1]$ implies $\tilde{\pi} \models \psi[r]$. Analogously, a state property $\Phi[r]$ is called increasing if $s \models \Phi[r]$ implies $s \models \Phi[r{+}1]$. Examples for increasing state properties are $\exists \mathrm{P}_{>p}(\varphi[r])$, $\forall \mathrm{P}_{>p}(\varphi[r])$, $\exists \mathrm{P}_{<p}(\psi[r])$ and $\forall \mathrm{P}_{<p}(\psi[r])$. Decreasing state properties are defined accordingly.

Sub-MDPs, end components. We use the notion *sub-MDP* of $\mathcal{M}$ for any pair $(T, \mathfrak{A})$ where $T \subseteq S$ and $\mathfrak{A} : T \to 2^{\mathit{Act}}$ such that for all $t \in T$: (1) $\mathfrak{A}(t) \subseteq \mathit{Act}(t)$ and (2) if $\alpha \in \mathfrak{A}(t)$ and $P(t, \alpha, t') > 0$ then $t' \in T$. An *end component* of $\mathcal{M}$ is a sub-MDP $(T, \mathfrak{A})$ of $\mathcal{M}$ where $\mathfrak{A}(t)$ is nonempty for all $t \in T$ and the underlying directed graph with node set T and the edge relation $t \to t'$ iff $P(t, \alpha, t') > 0$ for some $\alpha \in \mathfrak{A}(t)$ is strongly connected. An end component is said to be *maximal* if it is not contained in any other end component.

3 Quantiles

As stated in the introduction, quantiles in MDPs can be defined for arbitrary objectives given by increasing or decreasing parameterized state properties. We now provide general definitions for quantiles in MDPs where the state properties impose either a probability or an expectation constraint, and identify the instances for which we present algorithms in the next two sections.

Quantiles for probability objectives. Let $\mathcal{M} = (S, \mathit{Act}, P)$ be an MDP as in Sec. 2 and $\mathit{rew} : S \times \mathit{Act} \to \mathbb{N}$ a distinguished reward function in its reward structure. Given an increasing path property $\varphi[r]$ where parameter $r \in \mathbb{N}$ stands for some bound on the accumulated reward, we define the following types of *existential quantiles*, where $\psi[r] = \neg\varphi[r]$, $\trianglerighteq \in \{\geqslant, >\}$ and

$p \in [0,1] \cap \mathbb{Q}$:

$$
\begin{aligned}
\mathsf{Qu}_s(\,\exists\,\mathrm{P}_{\trianglerighteq p}(\varphi[?])\,) &= \min\ \{\, r \in \mathbb{N} \,:\, s \models \exists\,\mathrm{P}_{\trianglerighteq p}(\varphi[r])\,\} \\
&= \min\ \{\, r \in \mathbb{N} \,:\, \mathsf{Pr}^{\max}_s(\varphi[r]) \trianglerighteq p\,\} \\
\mathsf{Qu}_s(\,\exists\,\mathrm{P}_{\trianglerighteq p}(\psi[?])\,) &= \max\ \{\, r \in \mathbb{N} \,:\, s \models \exists\,\mathrm{P}_{\trianglerighteq p}(\psi[r])\,\} \\
&= \max\ \{\, r \in \mathbb{N} \,:\, \mathsf{Pr}^{\max}_s(\psi[r]) \trianglerighteq p\,\}
\end{aligned}
$$

Similarly, we can define the corresponding types of *universal quantiles*:

$$
\begin{aligned}
\mathsf{Qu}_s(\,\forall\,\mathrm{P}_{\trianglerighteq p}(\varphi[?])\,) &= \min\ \{\, r \in \mathbb{N} \,:\, \mathsf{Pr}^{\min}_s(\varphi[r]) \trianglerighteq p\,\} \\
\mathsf{Qu}_s(\,\forall\,\mathrm{P}_{\trianglerighteq p}(\psi[?])\,) &= \max\ \{\, r \in \mathbb{N} \,:\, \mathsf{Pr}^{\min}_s(\psi[r]) \trianglerighteq p\,\}
\end{aligned}
$$

From each of these quantiles we can derive three more quantiles by applying duality arguments, e.g., $\mathsf{Pr}^{\max}_s(\varphi[r]) = 1 - \mathsf{Pr}^{\min}_s(\psi[r])$, and the fact that $\min\{r \in \mathbb{N} : s \models \Phi[r]\}$ equals $\max\{r \in \mathbb{N} : s \not\models \Phi[r{-}1]\}$ when $\Phi[r]$ is an increasing state property. For example:

$$
\begin{aligned}
\min\{\, r \in \mathbb{N} \,:\, \mathsf{Pr}^{\max}_s(\varphi[r]) > p\,\} &= \min\ \{\, r \in \mathbb{N} \,:\, \mathsf{Pr}^{\min}_s(\psi[r]) < 1{-}p\,\} \\
&= \max\ \{\, r \in \mathbb{N} \,:\, \mathsf{Pr}^{\min}_s(\psi[r{-}1]) \geqslant 1{-}p\,\} \\
&= \max\ \{\, r \in \mathbb{N} \,:\, \mathsf{Pr}^{\max}_s(\varphi[r{-}1]) \leqslant p\,\}
\end{aligned}
$$

This observation yields groups of four quantiles that are derivable from each other. See [2] for the list of quantile dualities. For the above example we have:

$$
\begin{aligned}
\mathsf{Qu}_s(\,\exists\,\mathrm{P}_{>p}(\varphi[?])\,) &= \mathsf{Qu}_s(\,\forall\,\mathrm{P}_{<1-p}(\psi[?])\,) \\
&= \mathsf{Qu}_s(\,\forall\,\mathrm{P}_{\geqslant 1-p}(\psi[?])\,) + 1 \quad = \quad \mathsf{Qu}_s(\,\exists\,\mathrm{P}_{\leqslant p}(\varphi[?])\,) + 1
\end{aligned}
$$

The quantiles studied in [22] are obtained by considering $\varphi[r] = A\,\mathrm{U}^{\leqslant r}\,B$ and $\psi[r] = (\neg A)\,\mathrm{R}^{\leqslant r}(\neg B)$. Additionally, we address until-properties with lower reward bounds, i.e., $\varphi[r] = A\,\mathrm{U}^{\geqslant r}\,B$ and $\psi[r] = (\neg A)\,\mathrm{R}^{\geqslant r}(\neg B)$. To investigate the interplay of two reward functions (such as one for energy and one for utility) we also address path formulas where instead of the sets A and B, constraints for some other reward function are imposed. For instance:

$$
\lambda_{e,u} \;=\; \Diamond(\,(\mathit{energy} \leqslant e) \wedge (\mathit{utility} \geqslant u)\,),
$$

where $e, u \in \mathbb{N}$ and *energy* and *utility* stand for the accumulated reward along finite paths of reward functions $\mathit{erew} : S \times \mathit{Act} \to \mathbb{N}$ (for the energy) and $\mathit{urew} : S \times \mathit{Act} \to \mathbb{N}$ (for the utility). For an infinite path $\tilde{\pi}$, we have $\tilde{\pi} \models \lambda_{e,u}$ iff $\tilde{\pi}$ has a finite prefix π with $\mathit{erew}(\pi) \leqslant e$ and $\mathit{urew}(\pi) \geqslant u$. Likewise, $\lambda_{e,u}$ can be interpreted as an instance of an until-property with an upper or a lower reward bound. For fixed utility threshold u, the path property $\varphi[e] = \lambda_{e,u} = \Diamond^{\leqslant e}(\mathit{utility} \geqslant u)$ is increasing, while $\psi[u] = \lambda_{e,u} = \Diamond^{\geqslant u}(\mathit{energy} \leqslant e)$ is decreasing for fixed energy budget e. The task to compute the existential quantiles

$$
\begin{aligned}
\mathsf{Qu}_s(\,\exists\,\mathrm{P}_{>p}(\lambda_{?,u})\,) &= \min\ \{\, e \in \mathbb{N} \;:\; \mathsf{Pr}^{\max}_s(\lambda_{e,u}) > p\,\} \\
\mathsf{Qu}_s(\,\exists\,\mathrm{P}_{>p}(\lambda_{e,?})\,) &= \max\ \{\, u \in \mathbb{N} \;:\; \mathsf{Pr}^{\max}_s(\lambda_{e,u}) > p\,\}
\end{aligned}
$$

corresponds to the problem of constructing a scheduler that minimizes the energy ensuring that the achieved utility is at least u with probability $> p$ or to maximize the achieved degree of utility for a given energy budget e. Analogously, universal quantiles provide the corresponding information on the energy-utility characteristics in worst-case scenarios.

Quantiles for expectation objectives. We also consider quantiles where the objective is the minimal or maximal expected value of a random variable $f[r] : \mathit{IPaths} \to \mathbb{N} \cup \{\infty\}$. For instance, if $f[r]$ is increasing in r and θ some rational threshold, then an expectation quantile can be defined as the least $r \in \mathbb{N}$ such that the expected value of $f[r]$ is larger than θ for all or some scheduler(s). As an example for quantiles with expectation objectives, we consider a Boolean condition *cond* for finite paths and the random variable $f[e] = \mathit{utility}|_{\mathit{cond}} : \mathit{IPaths} \to \mathbb{N} \cup \{\infty\}$ that returns the utility value that is earned along finite paths where *cond* holds. Formally:

$$\mathit{utility}|_{\mathit{cond}}(\tilde{\pi}) \ = \ \sup\{\, \mathit{urew}(\mathit{pref}(\tilde{\pi}, k)) \ : \ k \in \mathbb{N},\, \mathit{pref}(\tilde{\pi}, k) \models \mathit{cond}\,\}$$

That is, if $\tilde{\pi}$ is an infinite path with $\tilde{\pi} \models \Diamond \mathit{cond}$ (i.e., $\mathit{pref}(\tilde{\pi}, k) \models \mathit{cond}$ for some $k \in \mathbb{N}$) then $\mathit{utility}|_{\mathit{cond}}(\tilde{\pi}) = \mathit{urew}(\pi)$, where π is the longest prefix of $\tilde{\pi}$ with $\pi \models \mathit{cond}$. If $\tilde{\pi} \models \Box \mathit{cond}$ (i.e., $\mathit{pref}(\tilde{\pi}, k) \models \mathit{cond}$ for all $k \in \mathbb{N}$) then $\mathit{utility}|_{\mathit{cond}}(\tilde{\pi})$ can be finite or infinite, depending on whether there are infinitely many positions i with $\mathit{urew}(s_i, \alpha_i) > 0$. Given a scheduler $\mathfrak{S}$ and a state s in $\mathcal{M}$, the *expected utility* for condition *cond* is the expected value of the random variable $\mathit{utility}|_{\mathit{cond}}$ under the probability measure induced by $\mathfrak{S}$ and s:

$$\mathsf{ExpUtil}_s^{\mathfrak{S}}(\,\mathit{cond}\,) \ = \ \sum_{r \in \mathbb{N}} r \cdot \mathsf{Pr}_s^{\mathfrak{S}}\{\, \tilde{\pi} \in \mathit{IPaths} \ : \ \mathit{utility}|_{\mathit{cond}}(\tilde{\pi}) = r\,\}$$

Note that $\mathsf{ExpUtil}_s^{\mathfrak{S}}(\,\mathit{cond}\,) = \infty$ is possible if $\mathsf{Pr}_s^{\mathfrak{S}}(\,\Diamond\Box(\mathit{cond})\,) > 0$. We define

$$\mathsf{ExpUtil}_s^{\max}(\,\mathit{cond}\,) \ = \ \sup_{\mathfrak{S}} \ \mathsf{ExpUtil}_s^{\mathfrak{S}}(\,\mathit{cond}\,).$$

$\mathsf{ExpUtil}_s^{\min}(\mathit{cond})$ is defined accordingly, taking the infimum over all schedulers rather than the supremum. Expectation energy-utility quantiles can be formalized by dealing with conditions $\mathit{cond}[e]$ that are parameterized by some energy value $e \in \mathbb{N}$. Examples are the following quantiles that fix a lower bound u for the extremal expected degree of utility and ask to minimize the required energy:

$$\mathsf{Qu}_s(\,\exists\, \mathrm{ExpU}_{>u}(\,\mathit{energy} \leqslant ?\,)\,) \ = \ \min \ \{\, e \in \mathbb{N} \ : \ \mathsf{ExpUtil}_s^{\max}(\,\mathit{energy} \leqslant e\,) > u\,\}$$

$$\mathsf{Qu}_s(\,\forall\, \mathrm{ExpU}_{>u}(\,\mathit{energy} \leqslant ?\,)\,) \ = \ \min \ \{\, e \in \mathbb{N} \ : \ \mathsf{ExpUtil}_s^{\min}(\,\mathit{energy} \leqslant e\,) > u\,\}$$

where $\pi \models (\mathit{energy} \leqslant e)$ iff $\mathit{erew}(\pi) \leqslant e$. Analogous definitions can be provided for quantiles that ask to maximize the achieved utility if an upper bound e for the expected consumed energy is given.

$$
\begin{array}{lcll}
\multicolumn{4}{l}{\text{minimize} \sum\limits_{(s,i)\in S[r]} x_{s,i} \text{ where } S[r] = S \times \{0,1,\ldots,r\}, \text{ subject to}} \\
x_{s,i} & = & 0 & \text{if } s \not\models \exists(A \,\mathrm{U}\, B) \text{ and } 0 \leqslant i \leqslant r \\
x_{s,i} & = & 1 & \text{if } s \in B \text{ and } 0 \leqslant i \leqslant r \\
x_{s,i} & \geqslant & \sum\limits_{t\in S} P(s,\alpha,t)\cdot x_{t,i-rew(s,\alpha)} & \text{if } s \notin B,\ s \models \exists(A \,\mathrm{U}\, B) \text{ and } \alpha \in Act(s) \\
 & & & \text{such that } rew(s,\alpha) \leqslant i \leqslant r
\end{array}
$$

Figure 1: Linear program $\mathbb{LP}_r$ with the unique solution $p_{s,i} = \mathsf{Pr}_s^{\max}(A\,\mathrm{U}^{\leqslant i}\,B)$

4 Computing probability quantiles

We now present algorithms for the computation of the quantitative quantiles introduced in Sec. 3. We start in this section with quantiles where the objective is a constraint on the extremal probability for a reward-bounded until formula. As stated before, quantiles that refer to reward-bounded release formulas are dual and can be computed using the same techniques.

Recently, a linear-programming (LP) approach for computing quantiles for (constrained) reachability properties with upper reward bounds (briefly called $\mathrm{U}^{\leqslant ?}$-quantiles) in MDPs with state rewards has been suggested [22]. We first recall this approach for quantitative $\mathrm{U}^{\leqslant ?}$-quantiles (Sec. 4.1) and then provide an efficient computation scheme that relies on an iterative back-propagation procedure including several heuristics (Sec. 4.2). In Sec. 4.3, we briefly show how to adapt these methods for reachability properties with lower reward bounds.

4.1 Iterative linear-programming based approach

We recall the approach of [22], focusing on existential $\mathrm{U}^{\leqslant ?}$-quantiles with strict probability bounds. Other $\mathrm{U}^{\leqslant ?}$-quantiles can be treated similarly (see [22]).

The idea for computing $\mathsf{Qu}_s(\,\exists\mathrm{P}_{>p}(A\,\mathrm{U}^{\leqslant ?}\,B)\,)$ is to first apply standard methods for computing the maximum probability $p_s = \mathsf{Pr}_s^{\max}(A\,\mathrm{U}\,B)$ for the unbounded until formula $A\,\mathrm{U}\,B$. If p_s does not meet the probability bound p, i.e., $p_s \leqslant p$, the quantile is infinite for state s. For $p_s > p$, the idea of [22] is to compute the maximal probabilities $p_{s,r} = \mathsf{Pr}_s^{\max}(A\,\mathrm{U}^{\leqslant r}\,B)$ for increasing reward bound r, until $p_{s,r} > p$. For this purpose, [22] provides an LP with variables $x_{s,i}$ for $(s,i) \in S[r] = S \times \{0,1,\ldots,r\}$ and the unique solution $(p_{s,i})_{(s,i)\in S[r]}$, where $p_{s,i} = \mathsf{Pr}_s^{\max}(\,A\,\mathrm{U}^{\leqslant i}\,B\,)$. Fig. 1 shows the LP of [22], adapted for the case of state-action rewards (rather than state rewards). This LP-based computation scheme can be solved in exponential time, as shown in [22] by establishing an upper bound r_{max} for the smallest (finite) quantile. A naïve approach thus could first compute r_{max}, generate the LP with variables $x_{s,i}$ for $(s,i) \in S[r_{\mathsf{max}}]$ and then use general-purpose linear- or dynamic-programming techniques to solve the constructed LP (e.g., the Simplex algorithm, ellipsoid methods or value or policy iteration). However, since the upper bound r_{max} is exponential in the size of $\mathcal{M}$ and depends on the number of states in $\mathcal{M}$, the transition probabilities and rewards in $\mathcal{M}$ and the probability bound p, this approach turns out to be intractable when $\mathcal{M}$ or the reward values are large.

4.2 Back-propagation approach

The main bottleneck of the LP approach for computing quantitative quantiles is the possibly exponential size of the LP. We propose an iterative approach that computes the values $p_{s,i} = \mathsf{Pr}_s^{\max}(A \,\mathrm{U}^{\leqslant i}\, B)$ successively for $i = 0, 1, 2, \ldots$ by decomposing the LP in Fig. 1 into smaller ones and propagating already computed values as much as possible. Due to the reuse of already computed values, we call this approach *back-propagation (BP) approach*.

Given that the solution $(p_{s,j})_{0 \leqslant j < i}$ for $\mathbb{LP}_{i-1}$ is known when considering $\mathbb{LP}_i$, the constraints for variable $x_{s,i}$ in the third case of Fig. 1 (i.e., if $s \notin B$, $s \models \exists(A \,\mathrm{U}\, B)$ and $\alpha \in Act(s)$) can be rewritten as follows:

$$\begin{array}{lll} x_{s,i} & \geqslant & c_{s,i} \stackrel{\text{def}}{=} \max\{\sum\limits_{t \in S} P(s,\alpha,t) \cdot p_{t,i-rew(s,\alpha)} \;:\; \alpha \in Act(s), rew(s,\alpha) > 0\} \\ x_{s,i} & \geqslant & \sum_{t \in S} P(s,\alpha,t) \cdot x_{t,i} \quad \text{if } rew(s,\alpha) = 0 \end{array}$$

We can now use standard methods to solve $\mathbb{LP}'_i$ with variables $(x_{s,i})_{s \in S}$ consisting of the above linear constraints together with the terminal cases $x_{s,i} = 0$ if $s \not\models \exists(A \,\mathrm{U}\, B)$ and $x_{s,i} = 1$ if $s \in B$, where the objective is to "minimize $\sum_{s \in S} x_{s,i}$". $\mathbb{LP}'_i$ has indeed a unique solution which agrees with the (unique) solution $(p_{s,i})_{s \in S}$ of $\mathbb{LP}_i$ for the variables $x_{s,i}$.

Suppose the task is to compute $q_s = \mathsf{Qu}_s(\exists \mathrm{P}_{>p}(A \,\mathrm{U}^{\leqslant ?}\, B))$ for all states s. Let $n = |S|$, $m = \sum_{s \in S} |Act(s)|$ and z be the number of state-action pairs (s,α) for which $s \in S$, $\alpha \in Act(s)$ and $rew(s,\alpha) = 0$. Then, with the proposed back-propagation approach, $(q_s)_{s \in S}$ is obtained by first computing $\mathsf{Pr}_s^{\max}(A \,\mathrm{U}\, B)$ for all states s (which can be done in time polynomial in the size of $\mathcal{M}$ [7, 5] and serves to identify the states $s \in S$ where $q_s = \infty$) and then solving the LPs $\mathbb{LP}'_0, \mathbb{LP}'_1, \ldots, \mathbb{LP}'_r$ (where $r \in \max\{q_s : \mathsf{Pr}_s^{\max}(A \,\mathrm{U}\, B) > p\}$) with n variables and $z + |S|$ linear constraints each.

Reward window. To reduce the memory requirements, we can use the observation that the constants $c_{s,i}$ in $\mathbb{LP}'_i$ are obtained from the values $p_{t,i-rew(s,\alpha)}$ where $\alpha \in Act(s)$ and $rew(s,\alpha) > 0$. As a consequence, the solution $(p_{t,i})_{t \in S}$ for $\mathbb{LP}'_i$ can be discarded as soon as $\mathbb{LP}'_{i+w}$ has been solved for the maximal reward value $w = \max\{rew(s,\alpha) : s \in S, \alpha \in Act(s)\}$ in $\mathcal{M}$. A further improvement considers the maximum reward of all incoming transitions per state. That is, the value of $p_{t,i}$ is not needed any more as soon as $\mathbb{LP}'_{i+w}$ has been solved where w equals the maximal reward of the state-action pairs (s,α) with $P(s,\alpha,t) > 0$.

Linear programs for zero-reward sub-MDP. The back-propagation approach can yield a major speed-up compared to the naïve approach with a single LP. However, if the number of state-action pairs with zero reward is large compared to the full set of actions in S, $\mathbb{LP}'_i$ needs still to be solved for several i. The idea then is to decompose $\mathbb{LP}'_i$ and treat the sub-LPs in a specific order. Let $\mathcal{G}$ be the directed graph with node set S and the edge relation $\rightarrow \subseteq S \times S$ given by $s \rightarrow t$ iff $P(s,\alpha,t) > 0$ for some action $\alpha \in Act(s)$ with $rew(s,\alpha) = 0$. Applying standard graph algorithms, we compute the strongly connected components in $\mathcal{G}$ and a topological sorting $C_1, \ldots, C_k$ for them. Then the SCCs $C_1, \ldots, C_k$ are the finest partition of S such that: if $s \in C_h$, $t \in C_j$, $P(s,\alpha,t) > 0$ and $rew(s,\alpha) = 0$, then $h \leqslant j$. Thus, we can decompose $\mathbb{LP}'_i$ into LPs $\mathbb{LP}'_{i,1}, \ldots, \mathbb{LP}'_{i,k}$, where $\mathbb{LP}'_{i,h}$ consists of the linear constraints $x_{s,i} \geqslant c_{s,i}$ and

$$x_{s,i} \geqslant \sum_{t \in C_h} P(s,\alpha,t) \cdot x_{t,i} + \sum_{u \in C_{>h}} P(s,\alpha,u) \cdot p_{u,i}$$

for $s \in C_h$, $\alpha \in Act(s)$, $rew(s,\alpha) = 0$. Here, $C_{>h} = C_{h+1} \cup \ldots \cup C_k$ and $(p_{u,i})_{u \in C_j}$ denotes the solutions of $\mathbb{LP}'_{i,j}$. The objective of $\mathbb{LP}'_{i,h}$ is to minimize the sum $\sum_{s \in C_h} x_{s,i}$.

Assuming that the sub-MDP $\mathcal{M}|_{rew=0}$ of $\mathcal{M}$ resulting by removing all actions α from $Act(s)$ with $rew(s,\alpha) > 0$ is acyclic, no LP has to be solved within our approach. In this case, the sets $C_1, \ldots, C_k$ are singletons, say $C_h = \{s_h\}$, and the solution $(p_{s,i})_{s \in S}$ is obtained directly when processing the states in reversed topological order $s_k, s_{k-1}, \ldots, s_1$.

Other improvements. Several other heuristics can be integrated to speed up the computation time or to decrease the memory requirements. For instance, zero-reward self-loops can be removed by a quantile-preserving transformation $\mathcal{M} \rightsquigarrow \mathcal{M}'$. The MDP $\mathcal{M}'$ has the same state space S as $\mathcal{M}$ and the same rewards for all state-action pairs. The transition probability function P' of $\mathcal{M}'$ is given by $P'(s,\alpha,t) = P(s,\alpha,t)/(1 - P(s,\alpha,s))$ if $rew(s,\alpha) = 0$, $t \neq s$ and $0 < P(s,\alpha,s) < 1$ and $P'(s,\alpha,t) = P(s,\alpha,t)$ in all other cases (see [2]). Another heuristic, which is however not yet realized in our implementation, is the aggregation method proposed in [8]. This approach permits to collapse all states belonging to the same maximal end components in the sub-MDP $\mathcal{M}|_{rew=0}$ into a single state.

4.3 Lower reward bounds

The approach for computing $\mathrm{U}^{\leqslant ?}$-quantiles can be adapted to compute quantiles for (constrained) reachability formulas with lower reward bounds, i.e., $A\,\mathrm{U}^{\geqslant ?}\,B$. For simplicity, we sketch only the treatment of reachability ($\Diamond^{\geqslant ?}B$) with a lower reward bound. More details and proofs can be found in [2]. We start with the universal quantile:

$$\mathsf{Qu}_s(\,\forall\, \mathrm{P}_{<p}(\Diamond^{\geqslant ?}B)\,) \;=\; \min\{\, r \in \mathbb{N} \,:\, \mathsf{Pr}^{\max}_s(\,\Diamond^{\geqslant r}B\,) < p\,\}$$

Clearly, if $\mathsf{Pr}^{\max}_s(\Diamond B) < p$ then the quantile for state s is 0. Furthermore:

$$\mathsf{Qu}_s(\,\forall\, \mathrm{P}_{<p}(\Diamond^{\geqslant ?}B)\,) = \infty \quad \text{iff} \quad \mathsf{Pr}^{\max}_s(\,\Diamond(C \wedge \Diamond B)\,) \geqslant p,$$

where C consists of all states t that are contained in a maximal end component $(T,\mathfrak{A})$ with $rew(t',\alpha) > 0$ for some state $t' \in T$ and an action $\alpha \in \mathfrak{A}(t')$. Intuitively, when entering C one can stay in C until the accumulated reward is greater or equal than r, before entering B. Otherwise, we apply the same idea as before and compute the values $p_{s,r} = \mathsf{Pr}^{\max}_s(\Diamond^{\geqslant r}B)$ for increasing r until $p_{s,r} < p$. The values $p_{s,r}$ are obtained as the unique solution of the following LP with variables $x_{s,i}$ for $(s,i) \in S\,[r]$ and the following constraints for $s \in S$ and $1 \leqslant i \leqslant r$:

$$x_{s,0} \;=\; \mathsf{Pr}^{\max}_s(\,\Diamond B\,)$$

$$x_{s,i} \;\geqslant\; 0$$

$$x_{s,i} \;\geqslant\; \sum_{t \in S} P(s,\alpha,t) \cdot x_{t,\ell} \qquad \text{if } \alpha \in Act(s) \text{ and } \ell = \max\{0, i - rew(s,\alpha)\}$$

The objective is to minimize $\sum_{(s,i)\in S[r]} x_{s,i}$. To speed up the computation, one can add the following constraints: $x_{s,i} = 1$ if $\mathsf{Pr}^{\max}_s(\,\Diamond(C \wedge \Diamond B)\,) = 1$ for $s \in S$.
The existential quantile

$$\mathsf{Qu}_s(\,\exists\mathrm{P}_{<p}(\Diamond^{\geqslant ?}B)\,) \;=\; \min\{\,r\in\mathbb{N} \,:\, \mathsf{Pr}^{\min}_s(\,\Diamond^{\geqslant r}B\,) < p\}$$

can then be computed by an analogous approach, using the fact that the values $p_{s,r} = \mathsf{Pr}^{\min}_s(\,\Diamond^{\geqslant r}B\,)$ are the greatest solutions in $[0,1]$ of the linear constraints

$$\begin{array}{lcll} x_{s,0} & = & \mathsf{Pr}^{\min}_s(\,\Diamond B\,) & \\ x_{s,i} & = & 0 & \text{if } \mathsf{Pr}^{\min}_s(\Diamond B) = 0 \text{ and } i \geqslant 1 \\ x_{s,i} & \leqslant & \sum\limits_{t\in S} P(s,\alpha,t)\cdot x_{t,\ell} & \text{if } \mathsf{Pr}^{\min}_s(\Diamond B) > 0,\, i \geqslant 1,\, \alpha\in Act(s) \\ & & & \text{and } \ell = \max\{0, i - rew(s,\alpha)\}. \end{array}$$

Then, $\mathsf{Qu}_s(\,\exists\mathrm{P}_{<p}(\Diamond^{\geqslant ?}B)\,) = \infty$ iff $\mathsf{Pr}^{\min}_s(\,\Box\Diamond B \,\wedge\, \Box\Diamond posR\,) \geqslant p$, where $posR \subseteq S \times Act$ is the set of state-action pairs (s,α) with $rew(s,\alpha) > 0$. Again, one could add the following constraints: $x_{s,i} = 1$ if $\mathsf{Pr}^{\min}_s(\Box\Diamond B \,\wedge\, \Box\Diamond posR) = 1$ for $s \in S$. Obviously, the back-propagation approach (cf. Sec. 4.2) is applicable for the existential and universal quantiles with lower bounds as well.

4.4 Energy-utility quantiles

The energy-utility quantile $\mathsf{Qu}_s(\,\exists\mathrm{P}_{>p}(\lambda_{?,u})\,)$ as introduced in Sec. 3 can be computed using the same techniques as explained for quantiles of the form $\mathsf{Qu}_s(\,\exists\mathrm{P}_{>p}(\Diamond^{\leqslant ?}B)\,)$. For this purpose, we might use an automaton $\mathcal{U}_u$ with states $q_0, q_1, \ldots, q_{u-1}, q_u$ representing the accumulated utility value. The goal state q_u represents that the achieved utility is at least u. The transitions of $\mathcal{U}_u$ are given by $q_i \to q_j$ for $j \geqslant i$. We put $\mathcal{M}$ and $\mathcal{U}_u$ in parallel to obtain an MDP $\mathcal{M}\otimes\mathcal{U}_u$ with a single reward function for the energy and synchronous transitions that capture the meaning of $\mathcal{U}_u$'s states. Formally, $\mathcal{M}\otimes\mathcal{U}_u = (S\times\{q_0,\ldots,q_u\}, Act, P')$ where

$$P'(\langle s,q_i\rangle,\alpha,\langle t,q_j\rangle) \;=\; P(s,\alpha,t) \quad \text{if } j = \min\{u, i + urew(s,\alpha)\}$$

and $P'(\cdot) = 0$ in all other cases. The reward structure of $\mathcal{M}\otimes\mathcal{U}_u$ consists of the energy reward function $erew$ lifted to the product. That is, we deal with the reward function $erew'$ for $\mathcal{M}\otimes\mathcal{U}_u$ given by $erew'(\langle s,q_i\rangle,\alpha) = erew(s,\alpha)$ for all $s\in S$, $0\leqslant i\leqslant u$ and $\alpha\in Act$. With $B = S\times\{q_u\}$, we then have

$$\mathsf{Pr}^{\max}_{\mathcal{M},s}(\,\Diamond(\,(energy \leqslant e)\wedge(utility \geqslant u)\,)\,) \;=\; \mathsf{Pr}^{\max}_{\mathcal{M}\otimes\mathcal{U}_u,\langle s,q_0\rangle}(\,\Diamond^{\leqslant e}B\,)$$

and therefore $\mathsf{Qu}^{\mathcal{M}}_s(\,\exists\mathrm{P}_{>p}(\lambda_{?,u})\,) = \mathsf{Qu}^{\mathcal{M}\otimes\mathcal{U}_u}_{\langle s,q_0\rangle}(\,\exists\mathrm{P}_{>p}(\,\Diamond^{\leqslant ?}B\,)\,)$.

The quantile $\mathsf{Qu}_s(\,\exists\mathrm{P}_{>p}(\lambda_{e,?})\,)$ is computable by an analogous automata-based approach, but now using the LP approach suggested for lower reward bounds (Sec. 4.3). Various other energy-utility quantiles can be computed using reductions to the case of reward-bounded until formulas or derived path properties. It is obvious that an analogous automata-based approach is applicable for quantiles where the objective is a probability constraint on path properties of the form $\Diamond((rew \bowtie r)\wedge\kappa)$, where κ is a Boolean combination of constraints of the form $rew_i \bowtie_i r_i$ for multiple reward functions $rew_1,\ldots,rew_k$ (other than rew).

5 Computing expectation quantiles

We now discuss how to compute the expectation quantiles in MDPs with two reward functions *erew* and *urew* for modeling the energy requirements and the achieved utility (see Sec. 3). Let us exemplify the approach computing

$$E_s^{\exists} \;=\; \mathsf{Qu}_s(\,\exists\,\mathrm{ExpU}_{>u}(\mathit{energy} \leqslant ?)\,) \text{ and } E_s^{\forall} \;=\; \mathsf{Qu}_s(\,\forall\,\mathrm{ExpU}_{>u}(\mathit{energy} \leqslant ?)\,).$$

Using known results for standard MDPs, we obtain that $\mathsf{ExpUtil}_s^{\max}(\mathit{energy} \leqslant e)$ is finite, provided that $\mathsf{Pr}_s^{\min}(\,\Diamond(\mathit{energy} > e)\,) = 1$. If, however, $\mathcal{M}$ contains end components where all the state-action pairs have zero energy reward then $\mathsf{Pr}_s^{\min}(\,\Diamond(\mathit{energy} > e)\,) < 1$ and $\mathsf{ExpUtil}_s^{\max}(\mathit{energy} \leqslant e) = \infty$ is possible.

Let us first make the simplifying assumption that all end components are both energy- and utility-divergent, i.e., whenever $(T, \mathfrak{A})$ is an end component of $\mathcal{M}$ then there exist state-action pairs (t, α) and (v, β) with $t, v \in T$ and $\alpha \in \mathfrak{A}(t)$, $\beta \in \mathfrak{A}(v)$ such that $\mathit{erew}(t, \alpha)$ and $\mathit{urew}(v, \beta)$ are positive. This assumption yields that $\mathsf{Pr}_s^{\min}(\,\Diamond(\mathit{energy} > e)\,) = 1$ and hence, $\mathsf{ExpUtil}_s^{\max}(\,\mathit{energy} \leqslant e\,)$ and $\mathsf{ExpUtil}_s^{\min}(\,\mathit{energy} \leqslant e\,)$ are finite for all states $s \in S$ and all energy bounds $e \in \mathbb{N}$. Moreover, $\lim_{e\to\infty} \mathsf{ExpUtil}_s^{\mathfrak{S}}(\mathit{energy} \leqslant e) \;=\; \infty$ for each scheduler $\mathfrak{S}$. This yields the finiteness of the expectation quantiles $E_s^{\exists}$ and $E_s^{\forall}$. The computation of $E_s^{\exists}$ and $E_s^{\forall}$ can be carried out using an iterative approach as for probability quantiles. For $E_s^{\exists}$, we compute iteratively the values $u_{s,e} = \mathsf{ExpUtil}_s^{\min}(\mathit{energy} \leqslant e)$ until $u_{s,e} > u$, in which case $E_s^{\exists} = e$. It remains to explain how to compute $u_{s,e}$. Again, we can use an LP-based approach and characterize the vector $(u_{s,i})_{(s,i)\in S[e]}$ as the unique solution of the LP with variables $x_{s,i}$ for $(s, i) \in S[e] = S \times \{0, 1, \ldots, e\}$ and the objective to maximize the sum of the $x_{s,i}$'s subject to:

$$x_{s,i} \;\leqslant\; \mathit{urew}(s, \alpha) + \sum_{t\in S} P(s, \alpha, t) \cdot x_{t,i-\mathit{erew}(s,\alpha)}$$

if $\alpha \in \mathit{Act}(s)$ and $\mathit{erew}(s, \alpha) \leqslant i \leqslant e$. For computing $E_s^{\forall}$, the values $v_{s,e} = \mathsf{ExpUtil}_s^{\max}(\mathit{energy} \leqslant e)$ can be computed by a similar schema, using the fact that the vector $(v_{s,i})_{(s,i)\in S[e]}$ is the least solution in $[0, 1]^{S[e]}$ of the linear constraints

$$x_{s,i} \;\geqslant\; \mathit{urew}(s, \alpha) + \sum_{t\in S} P(s, \alpha, t) \cdot x_{t,i-\mathit{erew}(s,\alpha)}$$

if $\alpha \in \mathit{Act}(s)$ and $\mathit{erew}(s, \alpha) \leqslant i \leqslant e$. Obviously, the back-propagation approach is applicable as well.

The computation of expectation quantiles for the general case, where no assumptions on the end components are imposed, are detailed in [2]. Basically, this computation relies on an analogous LP approach, but requires a preprocessing step to identify the states where $\mathsf{ExpUtil}_s^{\max}(\mathit{energy} \leqslant e) = \infty$, respectively $\mathsf{ExpUtil}_s^{\min}(\mathit{energy} \leqslant e) = \infty$ and computing those states where the quantile is infinite. The main feature for this preprocessing is an analysis of end components, similar as in [10, 12].

6 Implementation and case studies

In this section, we deal with our implementation of the algorithms for computing $U^{\leqslant ?}$-quantiles presented in Sec. 4.1 and 4.2 and demonstrate its usability within case studies. Our implementation relies on the computation of extremal probabilities for upper *reward-bounded* until properties on top of the explicit engine of the prominent probabilistic model checker PRISM version 4.1 [14], which have not yet been supported within PRISM so far. We compute quantiles either by solving the LP of [22] (see Fig. 1) directly using the LP-solver LPSOLVE[1] or with our back-propagation approach (BP). Our first case study is taken from PRISM's benchmark suite [17], showing the applicability of our implementation on relatively small models and compare the performance of the LP and BP approach. Then, we turn to computing energy-utility quantiles for an energy-aware job-scheduling protocol. All calculations were carried out on a computer with two Intel E5-2680 8-core CPUs at 2.70 GHz with 384GB of RAM. More detailed information and further case studies can be found in [2].

Self-stabilization. The self-stabilizing protocol by Israeli and Jalfon is modeled[2] as an MDP for N equal processes organized in a ring, each having a token at the beginning and aiming to randomly send and receive tokens until the ring is in a stable state, i.e., only one process has a token. We used our quantile algorithms to compute the minimal number of steps required for reaching a stable state with probability of at least p for some schedulers (existential quantile) or all schedulers (universal quantile). The latter problem also has been answered in the referred

Table 1: Results for randomized self-stabilizing (existential and universal quantile)

		model		existential quantile			universal quantile		
N	p	states	build	result	LP	BP	result	LP	BP
10	0.1	1,023	0.24s	18	118.38s	0.03s	26	403.36s	0.16s
	0.5	"	"	38	1,066.64s	0.05s	43	1,388.15s	0.09s
	0.99	"	"	117	11,552.55s	0.14s	130	19,794.61s	0.15s
15	0.1	32,767	1.56s	42	timeout	1.85s	61	timeout	3.78s
	0.5	"	"	89	timeout	3.85s	100	timeout	4.10s
	0.99	"	"	270	timeout	11.42s	305	timeout	12.18s

PRISM case study, but by iteratively increasing the step bound until the probability bound p was met. Our approach is more elegant by implicitly computing the probability values and answering only one (quantile) query. Table 1 shows our results for the LP and BP approach, with a timeout of 12 hours. The time for BP covers the entire computation of the quantile value r. For LP, we report the time for solving the linear program $\mathbb{LP}_r$. As it can be seen, the LP approach turns out to be infeasible already for relatively small models, whereas the BP implementation performs well. Table 1 also reveals that especially within the LP approach the time spent for evaluating the quantile increases significantly when the probability bound p is high (and hence, also the

[1] http://lpsolve.sourceforge.net, we used version 5.5.2, presolving deactivated
[2] http://www.prismmodelchecker.org/casestudies/self-stabilisation.php#ij

Table 2: Results for energy-aware job scheduling (quantiles $e_{\min}$ and $u_{\max}$)

		model		quantile $e_{\min}$				model		quantile $u_{\max}$	
N	p	states	build	result	time	N	p	states	build	result	time
4	0.1	368,521	14.67s	179	37.43s	4	0.1	872,410	14.47s	7	173.71s
	0.5	"	"	198	37.02s		0.5	"	"	7	173.22s
	0.99	"	"	225	42.69s		0.99	"	"	7	155.66s
5	0.1	6,079,533	377.95s	242	1,058.48s	5	0.1	3,049,471	65.69s	9	812.19s
	0.5	"	"	266	1,135.65s		0.5	"	"	9	812.93s
	0.99	"	"	301	1,261.89s		0.99	"	"	9	736.93s

quantile value is high).

Energy-aware job scheduling. We now turn to an energy-aware job-scheduling protocol modeled as an MDP, for which we compute energy-utility quantiles. Assume a system of N processes which need to enter a critical section in order to perform tasks, each within a given deadline. Access to the critical section is exclusively granted by a scheduler, which selects processes only if they have requested to enter. When a process states such a request, a deadline counter is set and decreased over time even if the process did not enter the critical section yet. Since computing a task also requires a certain amount of time in the critical section, deadlines can be exceeded. Utility is hence provided in terms of tasks finished without exceeding their deadline. Each process consumes energy, especially if it is in the critical section, and the global energy consumption equals the sum of energy consumed by all processes. Additional dependencies between utility and energy arise as the scheduler can activate a turbo mode for the critical section, doubling the computation speed but tripling energy consumption. As motivated in the introduction, we are now interested in the following energy-utility quantiles, both illustrating the trade-off between energy and utility w.r.t. several probability bounds p. We consider the quantile for the minimal energy $e_{\min}$ required to guarantee u successfully finished tasks, and the quantile for the maximal number $u_{\max}$ of tasks successfully finished by one process requiring not more than e energy. Our experiments solving these quantiles used the BP implementation with parameters u=N, e=$50{\cdot}N$. The results shown in Table 2 illustrate that even for large model sizes with millions of states, our implementation of the BP algorithm is feasible. As expected, none of the quantile computations for $e_{\min}$ and $u_{\max}$ finished within 12 hours when we used the LP approach instead of our BP implementation.

7 Conclusion

We introduced a general notion of (energy-utility) quantiles for MDPs and extended the LP schema from [22] to compute quantitative quantiles with lower and upper reward bounds, where the objective can be a probability constraint or a constraint on an expectation. We implemented a BP approach for quantitative quantiles with upper reward bounds, which can significantly speed up quantile computations, and demonstrated its performance by means of case studies.

Bibliography

[1] S. Andova, H. Hermanns, and J.-P. Katoen. Discrete-time rewards model-checked. In *FORMATS'03*, volume 2791 of *LNCS*, pages 88–104. Springer, 2003.

[2] C. Baier, M. Daum, C. Dubslaff, J. Klein, and S. Klüppelholz. Energy-Utility Quantiles. Technical report, TU Dresden, `http://wwwtcs.inf.tu-dresden.de/ALGI/PUB/NFM14/`, 2014.

[3] C. Baier, M. Daum, B. Engel, H. Härtig, J. Klein, S. Klüppelholz, S. Märcker, H. Tews, and M. Völp. Waiting for locks: How long does it usually take? In *FMICS'12*, volume 7437 of *LNCS*, pages 47–62. Springer, 2012.

[4] C. Baier, C. Dubslaff, J. Klein, S. Klüppelholz, and S. Wunderlich. Probabilistic Model Checking for Energy-Utility Analysis. *Festschrift*, 2014 (to appear).

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

[6] D. Bertsekas and J. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

[7] A. Bianco and L. de Alfaro. Model checking of probabilistic and non-deterministic systems. In *FSTTCS'95*, volume 1026 of *LNCS*, pages 499–513. Springer, 1995.

[8] F. Ciesinski, C. Baier, M. Größer, and J. Klein. Reduction techniques for model checking Markov decision processes. In *QEST'08*, pages 45–54. IEEE, 2008.

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

[10] L. de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In *CONCUR'99*, volume 1664 of *LNCS*, pages 66–81, 1999.

[11] 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.

[12] V. Forejt, M. Kwiatkowska, G. Norman, and D. Parker. Automated verification techniques for probabilistic systems. In *SFM'11*, volume 6659 of *LNCS*, pages 53–113. Springer, 2011.

[13] B. Haverkort. *Performance of Computer Communication Systems: A Model-Based Approach*. Wiley, 1998.

[14] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A tool for automatic verification of probabilistic systems. In *TACAS'06*, volume 3920 of *LNCS*, pages 441–444. Springer, 2006.

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

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

[17] M. Kwiatkowska, G. Norman, and D. Parker. The PRISM benchmark suite. In *QEST'12*. IEEE Computer Society, 2012.

[18] Y. Kwon and G. Agha. A Markov reward model for software reliability. In *IPDPS'07*, pages 1–6. IEEE, 2007.

[19] F. Laroussinie and J. Sproston. Model checking durational probabilistic systems. In *FOSSACS'05*, volume 3441 of *LNCS*, pages 140–154. Springer, 2005.

[20] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994.

[21] R. J. Serfling. *Approximation Theorems of Mathematical Statistics*. Wiley, 1980.

[22] M. Ummels and C. Baier. Computing quantiles in Markov reward models. In *FOSSACS'13*, volume 7794 of *LNCS*, pages 353–368. Springer, 2013.

MEALS Partner Abbreviations

SAU: Saarland University, D

RWT: RWTH Aachen University, D

TUD: Technische Universität Dresden, D

INR: Institut National de Recherche en Informatique et en Automatique, FR

IMP: Imperial College of Science, Technology and Medicine, UK

ULEIC: University of Leicester, UK

TUE: Technische Universiteit Eindhoven, NL

UNC: Universidad Nacional de Córdoba, AR

UBA: Universidad de Buenos Aires, AR

UNR: Universidad Nacional de Río Cuarto, AR

ITBA: Instituto Técnológico Buenos Aires, AR