Model Checking Durational Probabilistic Systems

(Extended abstract) ⋆

François Laroussinie[1] and Jeremy Sproston[2]

[1]Lab. Spécification & Verification, ENS Cachan – CNRS UMR 8643, France
[2]Dipartimento di Informatica, Università di Torino, 10149 Torino, Italy
fl@lsv.ens-cachan.fr, sproston@di.unito.it

Abstract. We consider model-checking algorithms for durational probabilistic systems, which are systems exhibiting nondeterministic, probabilistic and discrete-timed behaviour. We present two semantics for durational probabilistic systems, and show how formulae of the probabilistic and timed temporal logic PTCTL can be verified on such systems. We also address complexity issues, in particular identifying the cases in which model checking durational probabilistic systems is harder than verifying non-probabilistic durational systems.

1 Introduction

Model checking is an automatic method for guaranteeing that a mathematical model of a system satisfies a formula representing a desired property [7]. Many real-life systems, such as multimedia equipment, communication protocols, networks and fault-tolerant systems, exhibit *probabilistic* behaviour, leading to the study of *probabilistic model checking* of probabilistic and stochastic models [19, 13, 8, 5, 4, 3, 14]. Similarly, it is common to observe complex *real-time* behaviour in such systems. Model checking of discrete-time systems against properties of timed temporal logics, which can refer to the time elapsed along system behaviours, has been studied extensively in, for example, [11, 6, 16].

In this paper, we aim to study model-checking algorithms for discrete-time probabilistic systems, which we call *durational probabilistic systems*. Our starting point is the work of Hansson and Jonsson [13], which considered model checking for discrete-time Markov chains (in which transitions always take duration 1) against properties of a probabilistic, timed temporal logic, and that of de Alfaro [10], which extended the approach of Hansson and Jonsson to Markov decision processes in which transitions can be of duration 0 or of duration 1. We extend this previous work by considering systems in which state-to-state transitions take arbitrary, natural numbered durations, in the style of durational transition graphs [16, 17]. We present two semantics for durational probabilistic systems: the *continuous semantics* considers intermediate states as time elapses, whereas the *jump semantics* does not consider such states. In this paper, we restrict our attention to *strongly non-Zeno* durational probabilistic systems, in which positive durations elapse in all loops of the system.

⋆ Supported in part by MIUR-FIRB Perf.

The temporal logic that we use to describe properties of durational probabilistic systems is PTCTL (Probabilistic Timed Computation Tree Logic). The logic PTCTL includes operators that can refer to bounds on exact time, expected time, and the probability of the occurrence of events. For example, the property "a request is followed by a response within 5 time units with probability 0.99 or greater" can be expressed by the PTCTL property $request \rightarrow \mathbb{P}_{\geq 0.99}(\mathtt{true} \mathsf{U}_{\leq 5} response)$. Similarly, the property "the expected amount of time which elapses before reaching an alarm state is not more than 60" can be expressed as $\mathbb{D}_{\leq 60}(alarm)$. The logic PTCTL extends the probabilistic temporal logic PCTL [13, 5], the real-time temporal logic TCTL [1], and the performance-oriented logic of de Alfaro [10] (a similar logic has also been studied in the continuous-time setting [15]).

After introducing durational probabilistic systems and PTCTL in Section 2, we present model-checking algorithms for both of the aforementioned semantics in Section 3. The novelty of these algorithms is that their running time is independent of the timing constants used in the description of the durational probabilistic system, and their *program complexity* is polynomial. Instead, to apply the previous methods of de Alfaro, Hansson and Jonsson to durational probabilistic systems, we would have to model explicitly intermediate states as time passes (even for the jump semantics), hence resulting in a blow-up of the size of the state space. The presented algorithms are restricted to temporal modalities with upper or lower time bounds; we show in Section 4 that the problem of model checking durational probabilistic systems against PTCTL formulae in which exact time bounds are used (that is, of the form $= c$) is PSPACE-hard, even for "qualitative" probabilistic properties in which the probability thresholds refer to 0 or 1 only. We also show the NP-hardness and co-NP-hardness of model checking fully probabilistic durational systems against general "quantitative" probabilistic properties including arbitrary probability thresholds and upper time bounds (of the form $\leq c$). On the positive side, model checking qualitative probabilistic properties of fully probabilistic, strongly non-Zeno durational probabilistic systems is Δ_2^p-complete and PSPACE-complete for the jump and continuous semantics, respectively, and model checking qualitative properties excluding exact time bounds is in PSPACE for general strongly non-Zeno durational probabilistic systems with the jump semantics.

2 Durational Probabilistic Systems

2.1 Syntax of Durational Probabilistic Systems

Let AP be a countable set of atomic propositions, which we assume to be fixed throughout the remainder of the paper. Let $\mathcal{I}$ be the set of finite intervals over $\mathbb{N}$. Given a set X, $\mathsf{Dist}(X)$ denotes the set of discrete probability distributions over X.

Definition 1. *A* durational probabilistic system *(DPS)* $\mathcal{D} = (Q, q_{init}, D, L)$ *comprises a finite set of states* Q *with an initial state* $q_{init} \in Q$*; a finite durational probabilistic, nondeterministic transition relation* $D \subseteq Q \times \mathcal{I} \times \mathsf{Dist}(Q)$ *such that, for each state* $q \in Q$*, there exists at least one tuple* $(q, \_, \_) \in D$*; and a labelling function* $L : Q \rightarrow 2^{AP}$.

Intuitively, the behaviour of a durational probabilistic system comprises of repeatedly letting time pass then taking a state-to-state transition (which we sometimes call an *action transition*). The interval ρ of some $(q, \rho, \mu) \in D$ specifies the duration of the corresponding transition. On entry to a state $q \in Q$, there is a nondeterministic choice of a triple $(q, \rho, \mu) \in D$. Then the system chooses, again nondeterministically, the amount of time that elapses, where the chosen amount must belong to ρ. Finally, the system moves *probabilistically* to a next state $q' \in Q$ with probability $\mu(q')$.

The size $|\mathcal{D}|$ of $\mathcal{D}$ is $|Q| + |D|$ plus the size of the encoding of the timing constants and probabilities used in $\mathcal{D}$. The timing constants (lower and upper bounds of transitions' intervals) are written in binary, and where, for each $(q, \rho, \mu) \in D$, the values $\mu(q')$ are written as fixed-precision binary numbers.

Durational fully probabilistic systems. A *durational fully probabilistic system* (DFPS) is a DPS where there is exactly one tuple $(q, \rho, \_) \in D$ for any state q, and where ρ is a singleton. In such a system there is no non-deterministic choice.

Strong non-Zenoness. A DPS $\mathcal{D} = (Q, q_{init}, D, L)$ is *strongly non-Zeno* if, for each state $q \in Q$, there does not exist a sequence of transitions $(q_0, \rho_0, \mu_0)...(q_n, \rho_n, \mu_n)$ of $\mathcal{D}$ such that $q_0 = q$, $\mu_i(q_{i+1}) > 0$ for all $0 \leq i < n$, $\mu_n(q_0) > 0$, and ρ_i is of the form $[0; \_]$ for all $0 \leq i \leq n$. Note that this property can easily be checked for a DPS. The concept of strong non-Zenoness is taken from previous work for timed automata [18]. The algorithms and the complexity results we show in this paper only deal with strongly non-Zeno DPSs.

2.2 Semantics of Durational Probabilistic Systems

We give a formal semantics to durational probabilistic system in terms of *timed Markov decision processes*.

Definition 2. *A* timed Markov decision processes *(TMDP)* $\mathsf{M} = (S, s_{init}, \rightarrow, lab)$ *comprises a finite set of states* S *with an initial state* $s_{init} \in S$*; a finite timed probabilistic, nondeterministic transition relation* $\rightarrow \subseteq S \times \mathbb{N} \times \mathsf{Dist}(S)$ *such that, for each state* $s \in S$*, there exists at least one tuple* $(s, \_, \_) \in \rightarrow$*; and a labelling function* $lab : S \rightarrow 2^{AP}$.

A special case of a timed Markov decision process is a *timed Markov chain* (TMC), in which, for each state $s \in S$, there exists exactly one tuple $(s, \_, \_) \in \rightarrow$. The size of TMDPs and the notion of strong non-Zenoness are defined as for DPSs, because a TMDP can be regarded as a DPS for which the intervals labelling transitions are all singletons.

The transitions from state to state of a TMDP are performed in two steps: given that the current state is s, the first step concerns a nondeterministic selection of $(s, d, \nu) \in \rightarrow$, where d corresponds to the duration of the transition; the second step comprises a probabilistic choice, made according to the distribution ν, as to which state to make the transition to (that is, we make a transition to a state $s' \in S$ with probability $\nu(s')$). We often denote such a transition by $s \xrightarrow{d,\nu} s'$, and write $s \xrightarrow{d,\nu}$ to indicate that there exists $(s, d, \nu) \in \rightarrow$. If $s \xrightarrow{d,\nu} s'$ is such that $\nu(s') = 1$, then for simplicity we write $s \xrightarrow{d} s'$.

An infinite or finite *path* of the timed Markov decision process M is defined as an infinite or finite sequence of transitions, respectively, such that the target state of one transition is the source state of the next. We use $Path_{fin}$ to denote the set of finite paths of M, and $Path_{ful}$ the set of infinite paths of M. If ω is finite, we denote by $last(\omega)$ the last state of ω. For any path ω, let $\omega(i)$ be its $(i+1)$th state. Let $Path_{ful}(s)$ refer to the set of infinite paths commencing in state $s \in S$. For an infinite path $\omega = s_0 \xrightarrow{d_0,\nu_0} s_1 \xrightarrow{d_1,\nu_1} \cdots$, the accumulated duration along ω until the ith state, denoted $Time(\omega, i)$, is equal to $\sum_{0 \leq j < i} d_j$.

In contrast to a path, which corresponds to a resolution of nondeterministic and probabilistic choice, an *adversary* represents a resolution of nondeterminism *only*. Formally, an adversary of a timed Markov decision process M is a function A mapping every finite path $\omega \in Path_{fin}$ to a transition $(last(\omega), d, \nu) \in \rightarrow$. Let Adv be the set of adversaries of M. For any adversary $A \in Adv$, let $Path^A_{ful}$ denote the set of infinite paths resulting from the choices of distributions of A, and let $Path^A_{ful}(s) = Path^A_{ful} \cap Path_{ful}(s)$. Then, for a state $s \in S$, we define the probability measure $Prob^A_s$ over $Path^A_{ful}(s)$ in the standard way [19].

Note that, by defining adversaries as functions from finite paths, we permit adversaries to be dependent on the history of the system. Hence, the choice made by an adversary at a certain point in system execution can depend on the sequence of states visited, the nondeterministic choices taken, and the time elapsed in each state, up to that point.

As for non-probabilistic systems [17], we can define several semantics of time for DPSs. Consider a transition of duration d between two DPS states q and q'. The first semantics, called the *jump* semantics, assumes that moving from q to q' takes d time units and that there are no intermediate states: if the system is in q at time t, then it is in q' at time $t + d$ and there is no position for time $t + 1 \ldots t + d - 1$. This semantics corresponds to a kind of cost or reward automata where every transition has a weight. We will also consider the *continuous* semantics, which involves waiting in $d - 1$ intermediate positions, each corresponding to the passage of one time unit, before performing the action transition and arriving in q'. This last semantics is close to the one used for timed automata and is generally more natural to model systems; for example, it is more convenient when considering parallel composition because time progresses smoothly.

Jump semantics. The *jump* semantics of a DPS $\mathcal{D} = (Q, q_{init}, D, L)$ is defined as the TMDP $\mathsf{M}_j(\mathcal{D}) = (S, s_{init}, \rightarrow, lab)$, where:

- $S = Q$ and $s_{init} = q_{init}$;
- $(s, d, \mu) \in \rightarrow$ if and only if there exists $(s, \rho, \mu) \in D$ and $d \in \rho$;
- $lab(s) = L(s)$ for all $s \in S$.

Continuous semantics. Let $\delta_{\max}(q)$ be the maximal delay possible in state q of a durational probabilistic system. The *continuous* semantics of a DPS $\mathcal{D} = (Q, q_{init}, D, L)$ is defined as the TMDP $\mathsf{M}_c(\mathcal{D}) = (S, s_{init}, \rightarrow, lab)$, where:

- $S = \{(q, i) \mid 0 \leq i < \delta_{\max}(q)\}$ and $s_{init} = (q_{init}, 0)$;

– $\rightarrow$ is the smallest set of transitions satisfying the following rules:
  • $(q, 0) \xrightarrow{0,\nu}$ if there exists $(q, \rho, \mu) \in D$ such that $0 \in \rho$, and where $\nu(q', 0) = \mu(q')$ for each $q' \in Q$;
  • $(q, i) \xrightarrow{1} (q, i+1)$ if $i + 1 < \delta_{\max}(q)$;
  • $(q, i) \xrightarrow{1,\nu}$ if there exists $(q, \rho, \mu) \in D$ such that $i+1 \in \rho$, and where $\nu(q', 0) = \mu(q')$ for each $q' \in Q$;
– for each $(q, i) \in S$, let $lab(q, i) = L(q)$.

Observe that the semantics of a DFPS is a TMC, and that the semantics of a strongly non-Zeno DPS is also strongly non-Zeno. The size of the transition relation of $\mathsf{M}_j(\mathcal{D})$ may be exponential in $|\mathcal{D}|$ because it is linearly-dependent on the magnitude of the timing constants (encoded in binary) of the DPS. However, the number of states of $\mathsf{M}_j(\mathcal{D})$ and $\mathcal{D}$ is the same. This contrasts with $\mathsf{M}_c(\mathcal{D})$, where the number of states *and* the number of transitions may be exponential in $|\mathcal{D}|$. Another difference between the semantics is that the TMDP $\mathsf{M}_c(\mathcal{D})$ only contains durations in $\{0, 1\}$.

2.3 Probabilistic Timed Temporal Logic

In this section, we recall how the branching-time temporal logic CTL can be extended with constraints on time, probability and expected time. First we recall the probabilistic temporal logic PCTL [13, 5], in which the standard universal and existential path quantifiers $\mathsf{A}\varphi$ and $\mathsf{E}\varphi$ of CTL are replaced with a probabilistic quantifier of the form $\mathbb{P}_{\bowtie\lambda}(\varphi)$, where φ is a formula interpreted over paths, $\bowtie \in \{<, \leq, \geq, >\}$ is a comparison operator and $\lambda \in [0; 1]$ is a probability. Timing constraints, expressed using subscripts on "until" path formulae (with the syntax $\mathsf{U}_{\sim c}$, where $\sim \in \{\leq, =, \geq\}$), were introduced in the temporal logics RTCTL [11] and TCTL [1]. Finally, an expected-time operator $\mathbb{D}_{\bowtie\zeta}(\Phi)$, where $\bowtie \in \{<, \leq, \geq, >\}$ is a comparison operator and $\zeta \in \mathbb{R}_{\geq 0}$ is a non-negative real, was studied in the discrete-time context by de Alfaro [10] and Andova et al. [2].

We combine the above mentioned temporal logics to obtain the temporal logic PTCTL (Probabilistic Timed Computation Tree Logic), which extends the identically-named logic of [15] with the "next" temporal modality and the expected-time operator.

Definition 3. *The formulae of* PTCTL *are given by the following grammar:*

$$\Phi ::= P \mid \Phi \wedge \Phi \mid \neg\Phi \mid \mathbb{P}_{\bowtie\lambda}(\mathsf{X}\Phi) \mid \mathbb{P}_{\bowtie\lambda}(\Phi \mathsf{U}_{\sim c}\Phi) \mid \mathbb{D}_{\bowtie\zeta}(\Phi)$$

where $P \in AP$ is an atomic proposition, $\bowtie \in \{<, \leq, \geq, >\}$, $\sim \in \{\leq, =, \geq\}$ are comparison operators, $\lambda \in [0; 1]$ is a probability, $c \in \mathbb{N}$ is a natural number, and $\zeta \in \mathbb{R}_{\geq 0}$ is a positive real.

We define PTCTL$[\leq, \geq]$ as the sub-logic of PTCTL in which subscripts of the form $= c$ are not allowed in "until" modalities $\mathsf{U}_{\sim c}$. The size $|\Phi|$ is defined in the standard way, with constants written in binary.

Given an infinite path ω of a TMDP and a PTCTL formula Φ, let $T_{\omega,\Phi} = \min\{i \mid \omega(i) \models \Phi\}$ be the index of the first state of ω which satisfies Φ, and let $T_{\omega,\Phi} = \infty$ if $\omega(i) \not\models \Phi$ for all $i \in \mathbb{N}$. Then, for a given adversary $A \in Adv$ and state $s \in S$ of the TMDP, we let $ExpectedTime_s^A(\Phi) = \mathrm{E}_s^A\{Time(\omega, T_{\omega,\Phi})\}$, where $\mathrm{E}_s^A\{\cdot\}$ is the expectation, defined in the standard way, with respect to the probability measure $Prob_s^A$.

Definition 4. *Given a TMDP* $\mathsf{M} = (S, s_{init}, \rightarrow, lab)$ *and a* PTCTL *formula* Φ*, we define the satisfaction relation* $\models_{\mathsf{M}}$ *of* PTCTL *as follows:* [1]

$$
\begin{array}{lll}
s \models_{\mathsf{M}} P & \textit{iff} \; P \in lab(s) \\
s \models_{\mathsf{M}} \Phi_1 \wedge \Phi_2 & \textit{iff} \; s \models_{\mathsf{M}} \Phi_1 \textit{ and } s \models_{\mathsf{M}} \Phi_2 \\
s \models_{\mathsf{M}} \neg\Phi & \textit{iff} \; s \not\models_{\mathsf{M}} \Phi \\
s \models_{\mathsf{M}} \mathbb{D}_{\bowtie\zeta}(\Phi) & \textit{iff} \; \mathit{ExpectedTime}_s^A(\Phi) \bowtie \zeta, \; \forall A \in Adv \\
s \models_{\mathsf{M}} \mathbb{P}_{\bowtie\lambda}(\varphi) & \textit{iff} \; \mathit{Prob}_s^A\{\omega \in \mathit{Path}_{ful}^A(s) \mid \omega \models_{\mathsf{M}} \varphi\} \bowtie \lambda, \; \forall A \in Adv \\
\omega \models_{\mathsf{M}} \mathsf{X}\Phi & \textit{iff} \; \omega(1) \models_{\mathsf{M}} \Phi \\
\omega \models_{\mathsf{M}} \Phi_1 \mathsf{U}_{\sim c} \Phi_2 & \textit{iff} \; \exists i \in \mathbb{N} \; s.t. \; \mathit{Time}(\omega, i) \sim c \;, \; \omega(i) \models_{\mathsf{M}} \Phi_2 \\
 & \quad \textit{and } \omega(j) \models_{\mathsf{M}} \Phi_1 \;\; \forall 0 \le j < i \;.
\end{array}
$$

Model checking. The model-checking problem for a PTCTL formula Φ and a TMDP M with initial state s_{init} is to decide whether $s_{init} \models_{\mathsf{M}} \Phi$, which we abbreviate to $\mathsf{M} \models \Phi$. The model-checking problem for Φ, a DPS $\mathcal{D}$ and a semantics $sem \in \{j, c\}$ is to decide whether $\mathsf{M}_{sem}(\mathcal{D}) \models \Phi$. The complexity results will be expressed in terms of the size $|\mathcal{D}| + |\Phi|$. However, we will also consider the *program complexity* where one fixes the formula and measures the complexity as a function of the size $|\mathcal{D}|$ only. As the system is assumed to be large whereas the formula is assumed to be small, the program complexity is often considered to be a more significant estimate of the feasibility of verification in practice.

3 Model Checking for Durational Probabilistic Systems

Our approach is to introduce in Section 3.1 a model-checking algorithm for strongly non-Zeno timed Markov decision processes, which will then be used in Section 3.2 as a basis for model-checking algorithms for durational probabilistic systems.

3.1 Model Checking Timed Markov Decision Processes

Although our model-checking algorithm for TMDPs presented below uses the analogous algorithm of de Alfaro [9] in order to verify the expected-time operator, the methods and complexities for the probabilistic, time-bounded operators are new, and, for strongly non-Zeno TMDPs, improve on previous results [13, 10] as their running time is not dependent on the magnitude of the time constants used in the transitions of the TMDP. More precisely, the previous methods are defined for systems in which the maximal time duration is 1, necessitating the modelling of longer time durations via intermediate states, hence blowing-up the size of the state space.

Before presenting the algorithm, we introduce some notation. The algorithm relies on computing a topological order on the states of the TMDP, so that reachability via 0 transitions is reflected in the order: for two states $s, s' \in S$, let $s \succ_0 s'$ if and only if there exists a transition $s' \xrightarrow{0,\nu}$ where $\nu(s) > 0$. Then we order the states in S

[1] When clear from the context, we omit the M subscript from $\models_{\mathsf{M}}$.

$\underline{\mathbb{P}_{\leq\lambda}(\Phi_1 \mathsf{U}_{\leq c} \Phi_2)}$:	for $i := 0$ to c
	  for $j := 0$ to n
	    if $s_j \models \Phi_2$ then let $f(s_j, i) := 1$
	    else
	      if $s_j \not\models \Phi_1 \vee \Phi_2$ then let $f(s_j, i) := 0$
	      else let $f(s_j, i) := \max_{(s_j,d,\nu)\in\rightarrow} \sum_{s'\in S} \nu(s') \cdot f(s', i-d)$

$\underline{\mathbb{P}_{\leq\lambda}(\Phi_1 \mathsf{U}_{=c} \Phi_2)}$:	for each $s \models \Phi_2$ let $f(s, 0) := 1$
	for $i := 0$ to c
	  for $j := 0$ to n
	    if $s_j \not\models \Phi_1 \vee \Phi_2$ then let $f(s_j, i) := 0$
	    else let $f(s_j, i) := \max_{(s_j,d,\nu)\in\rightarrow} \sum_{s'\in S} \nu(s') \cdot f(s', i-d)$

$\underline{\mathbb{P}_{\leq\lambda}(\Phi_1 \mathsf{U}_{\geq c} \Phi_2)}$:	for each $s \in S$
	  let $f(s, 0) := \sup_{A\in Adv} Prob_s^A\{\omega \in Path_{ful}^A(s) \mid \omega \models \Phi_1 \mathsf{U} \Phi_2\}$
	for $i := 0$ to c
	  for $j := 0$ to n
	    if $s_j \not\models \Phi_1 \vee \Phi_2$ then let $f(s_j, i) := 0$
	    else let $f(s_j, i) := \max_{(s_j,d,\nu)\in\rightarrow} \sum_{s'\in S} \nu(s') \cdot f(s', \max(0, i-d))$

Fig. 1. The algorithms for computing $\mathbb{P}_{\leq\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$

according to $\succ_0$ to obtain a sequence $s_0 s_1 ... s_n$ where $n = |S| - 1$, $s_{i+j} \not\succ_0 s_i$ for each $0 \leq i < n$, $1 \leq j \leq n - i$, and each state in S appears exactly once in the sequence. The fact that such a sequence $s_0 s_1 ... s_n$ exists follows from the fact that M is strongly non-Zeno. Computing the order can be done in time $O(|S| + |\xrightarrow{0}|)$ where $|\xrightarrow{0}| = \Sigma_{(s,0,\nu)\in\rightarrow} |\nu|$ and $|\nu| = |\{s' \mid \nu(s') > 0\}|$. In the algorithm below, we will always iterate over the states of the TMDP in such a way as to respect the topological order, in order to propagate the computed probabilities correctly through the states.

Proposition 1. *Let* $\mathsf{M} = (S, s_{init}, \rightarrow, lab)$ *be a strongly non-Zeno TMDP and* Φ *be a* PTCTL *formula in which the maximal constant in its time-bound subscripts is* c_{max}. *Deciding whether* $\mathsf{M} \models \Phi$ *can be done in time* $O(|\Phi| \cdot ((|S| \cdot |\rightarrow| \cdot c_{max}) + poly(|\mathsf{M}|)))$.

Proof. The cases for the atomic propositions, Boolean combinations and next formulae are standard, and therefore we concentrate on the model-checking algorithm for PTCTL formulae of the form $\mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$ and $\mathbb{D}_{\bowtie\zeta}(\Phi')$. We restrict our attention to the cases in which $\bowtie$ is $\leq$. The cases for $\geq$ are obtained directly by substituting $\min$ for $\max$, and $\inf$ for $\sup$ in the following procedures, and the cases for $\bowtie \in \{<, >\}$ follow similarly. We assume that arithmetical operations can be performed in constant time.

Until formulae. We consider three different procedures (see Figure 1) depending on the form of $\sim$. Recall that we use a topological order for enumerating the states $s_0 s_1 ... s_n$ in order to respect $\succ_0$.

In each of the procedures, a function of the form $f : S \times \mathbb{Z} \rightarrow [0; 1]$ is utilized, with the intuition that, for $0 \leq i \leq c$, the state s satisfies the path formula $\Phi_1 \mathsf{U}_{\sim i} \Phi_2$ with maximum probability $f(s,i)$. Naturally, the aim is to calculate $f(s,c)$ for each state $s \in S$. In each of the three cases, for each $i < 0$ and each $s \in S$, we assume that we have $f(s,i) = 0$. One can prove by induction over i that $f(s,i) = \sup_{A \in Adv} Prob_s^A\{\omega \in Path_{ful}^A(s) \mid \omega \models \Phi_1 \mathsf{U}_{\sim i} \Phi_2\}$ for each state $s \in S$ and each $0 \leq i \leq c$. Hence, we conclude that $s \models \mathbb{P}_{\leq \lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$ if and only if $f(s,c) \leq \lambda$. The complexity of the first two procedures, where $\sim$ is $\leq$ or $=$, is $O(c \cdot |S| \cdot |\rightarrow|)$.

When $\sim$ is $\geq$, our algorithm first requires that we compute, for each state $s \in S$, the probability $\sup_{A \in Adv} Prob_s^A\{\omega \in Path_{ful}^A(s) \mid \omega \models \Phi_1 \mathsf{U} \Phi_2\}$ (the maximum probability of satisfying the un-subscripted formula $\Phi_1 \mathsf{U} \Phi_2$). Following Bianco and de Alfaro [5], these probabilities can be computed in $O(poly(|\mathsf{M}|))$ time. Therefore, the complexity of the third procedure is $O((c \cdot |S| \cdot |\rightarrow|) + poly(|\mathsf{M}|))$.

Expected-time formulae. For formulae of the form $\mathbb{D}_{\bowtie \zeta}(\Phi')$, we can utilize the algorithm of de Alfaro [9] (TMDPs are a special case of de Alfaro's model), which reduces to a linear programming problem, with time complexity $poly(|\mathsf{M}|)$.

Overall complexity. We obtain an overall time complexity of $O(|\Phi| \cdot ((|S| \cdot |\rightarrow| \cdot c_{max}) + poly(|\mathsf{M}|)))$. Note that the time complexity can be expressed in terms of the maximum branching degree of the transitions of the TMDP. More precisely, if $b_{max} = \max_{(\_,\_,\nu) \in \rightarrow} |\{s \mid \nu(s) > 0\}|$ then we can write the complexity as $O(|\Phi| \cdot ((b_{max} \cdot |\rightarrow| \cdot c_{max}) + poly(|\mathsf{M}|)))$. □

3.2 Extension to Strongly Non-Zeno Durational Probabilistic Systems

We now show how the algorithms of Section 3.1 can be used to define PTCTL model-checking algorithms for DPSs. One idea would be to apply these algorithms directly to the semantic TMDP of a DPS; however, in both semantics, the corresponding TMDPs are exponential in the size of original DPS . We avoid this in the case of PTCTL$[\leq, \geq]$ by utilizing specific TMDP constructions for both of the semantics.

Proposition 2 (DPS with jump semantics). *Let $\mathcal{D} = (Q, q_{init}, D, L)$ be a strongly non-Zeno durational probabilistic system and Φ be a* PTCTL$[\leq, \geq]$ *formula in which the maximal constant in the subscripts is c_{max}. Deciding whether $\mathsf{M}_j(\mathcal{D}) \models \Phi$ can be done in time $O(|\Phi| \cdot ((|Q| \cdot |D| \cdot c_{max}) + poly(|\mathcal{D}|)))$.*

Proof (sketch). We define a TMDP $\mathsf{M}_j^r(\mathcal{D}) = (S, s_{init}, \rightarrow^r, lab)$ corresponding to a restricted version of the jump semantics of $\mathcal{D}$ where S, s_{init}, and lab are defined as for the standard jump semantics, and $(s, d, \mu) \in \rightarrow^r$ if and only if there exists $(s, [l;u], \mu) \in D$ and either $d = l$ or $d = u$. Then, for any state $s \in S$, we can show that $s \models_{\mathsf{M}_j(\mathcal{D})} \Phi$ if and only if $s \models_{\mathsf{M}_j^r(\mathcal{D})} \Phi$: the minimum and maximum probabilities and expectations depend only on the minimum and maximum durations on transitions. □

Proposition 3 (DPS with continuous semantics). *Let $\mathcal{D} = (Q, q_{init}, D, L)$ be a strongly non-Zeno durational probabilistic system and Φ be a* PTCTL$[\leq, \geq]$ *formula in which the maximal constant in the subscripts is c_{max}. Deciding whether $\mathsf{M}_c(\mathcal{D}) \models \Phi$ can be done in time $O((|\Phi|^3 \cdot |D|^3 \cdot c_{max}) + poly(|\Phi| \cdot |D| \cdot |\mathcal{D}|))$.*

Proof (sketch). We write the continuous semantics of $\mathcal{D}$ as $\mathsf{M}_c(\mathcal{D}) = (S, s_{init}, \rightarrow, lab)$. Our aim is to label every state (q, i) of $\mathsf{M}_c(\mathcal{D})$ with the set of subformulae of Φ

which it satisfies. For each state $q \in Q$, we construct a set $\mathsf{Sat}[q, \xi]$ of intervals such that $\alpha \in \mathsf{Sat}[q, \xi]$ if and only if $(q, \alpha) \models \xi$. For reasons of space, we explain only the general ideas behind the verification of subformulae Ψ of the form $\mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$ and $\mathbb{D}_{\bowtie\zeta}(\Phi')$. For this, we assume that we have already computed the sets $\mathsf{Sat}[\_, \_]$ for Φ_1, Φ_2 and Φ'.

As in Proposition 2, we construct a restricted TMDP which represents partially the states and transitions of $\mathsf{M}_c(\mathcal{D})$ but which will be sufficient for computing the sets $\mathsf{Sat}[q, \Psi]$. The size of the restricted TMDP will ensure a procedure running in time polynomial in $|\mathcal{D}|$.

For the interval $\rho = [l; u]$, let $\rho - 1$ be the interval $[\max(0, l-1); \max(0, u-1)]$. For each state $q \in Q$, we build the minimal set of intervals $\mathsf{Int}(q) = \bigcup_{j=1..k}[\alpha_j; \beta_j)$ such that:

- for any i, we have $i \in \mathsf{Int}(q)$ if and only if $i \in \mathsf{Sat}[q, \Phi_1] \cup \mathsf{Sat}[q, \Phi_2]$, and every interval of $\mathsf{Int}(q)$ verifies either $\Phi_1 \wedge \Phi_2$, $\Phi_1 \wedge \neg\Phi_2$ or $\neg\Phi_1 \wedge \Phi_2$;
- for any j, we have $\alpha_j < \beta_j$, and $\beta_j \leq \alpha_{j+1}$ if $j + 1 \leq k$;
- the intervals are *homogeneous for action transitions*: for any $(q, \rho, \_) \in D$, we have $[\alpha_j, \beta_j) \subseteq \rho - 1$ or $[\alpha_j, \beta_j) \cap \rho - 1 = \emptyset$;
- the interval $[0; 1)$ is treated separately: if $0 \in \mathsf{Sat}[q, \Phi_1] \cup \mathsf{Sat}[q, \Phi_2]$, then $[0; 1)$ is the first interval of $\mathsf{Int}(q)$.

Letting $D^q = \{(q, \_, \_) \mid (q, \_, \_) \in D\}$, we clearly have $|\mathsf{Int}(q)| \leq 2 \cdot (|\mathsf{Sat}[q, \Phi_1]| + |\mathsf{Sat}[q, \Phi_2]| + |D^q|) + 1$. Let ν be a *sub-distribution* on a set S if $\nu : S \to [0; 1]$ and $\sum_{s \in S} \nu(s) \leq 1$, and let $\mathsf{SubDist}(S)$ be the set of all sub-distributions on the set S. Next, we build $\mathsf{M}_I = (Q_I, \_, \rightarrow_I, lab_I)$, which is a variant of a TMDP in which sub-distributions may be used in addition to distributions. The set of states of M_I is $Q_I = \{(q, [\alpha; \beta)) \mid q \in Q \text{ and } [\alpha; \beta) \in \mathsf{Int}(q)\}$, and the set of timed probabilistic, nondeterministic transitions $\rightarrow_I \subseteq S \times \mathbb{N} \times \mathsf{SubDist}(S)$ is the smallest set defined as follows.

(Action transition) For any $(q, \rho, \mu) \in D$ and $[\alpha; \beta) \in \mathsf{Int}(q)$, if $[\alpha; \beta) \subseteq \rho - 1$, then:

if $[\alpha; \beta) = [0; 1)$**:** we have the transition $(q, [\alpha; \beta)) \xrightarrow{0,\nu}_I$ if $0 \in \rho$, and the transition $(q, [\alpha; \beta)) \xrightarrow{1,\nu}_I$ if $1 \in \rho$;

if $[\alpha; \beta) \neq [0; 1)$**:** we have the transitions $(q, [\alpha; \beta)) \xrightarrow{1,\nu}_I$ and $(q, [\alpha; \beta)) \xrightarrow{\beta-\alpha,\nu}_I$;

where $\nu \in \mathsf{SubDist}(Q_I)$ is the (sub-)distribution such that, for each $(q', [\alpha'; \beta')) \in Q_I$, we have:

$$\nu(q', [\alpha'; \beta')) = \begin{cases} \mu(q') & \text{if } [\alpha'; \beta') = [0; 1) \text{ and } [0; 1) \in \mathsf{Int}(q') \\ 0 & \text{otherwise.} \end{cases}$$

(Time successor) For any $[\alpha; \beta)$ and $[\alpha'; \beta')$ in $\mathsf{Int}(q)$, if $\beta = \alpha'$ then we have $(q, [\alpha, \beta)) \xrightarrow{\beta-\alpha}_I (q, [\alpha'; \beta'))$.

Finally, for each $(q, [\alpha; \beta)) \in Q_I$, we let $lab_I(q, [\alpha; \beta)) \subseteq \{\Phi_1, \Phi_2\}$ depending the inclusion of $[\alpha; \beta)$ w.r.t. $\mathsf{Sat}[q, \Phi_1]$ and $\mathsf{Sat}[q, \Phi_2]$.

The TMDP M_I has the following important property: for any state $(q, [\alpha; \beta))$ of M_I, we have that $(q, \alpha) \models_{\mathsf{M}_c(\mathcal{D})} \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$ if and only if $(q, [\alpha; \beta)) \models_{\mathsf{M}_I}$

$\mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$. This can be shown by using the same kind of arguments we used for proving Proposition 2.

Then using the above construction of M_I, we can apply the algorithm of Section 3.1 to decide, for each $(q, [\alpha; \beta)) \in Q_I$, whether $(q, \alpha) \models_{\mathsf{M}_c(\mathcal{D})} \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$ (the presence of sub-distributions does not affect the results of the algorithm). Now note that, for each function f considered in Section 3.1, we compute a value for each state $(q, [\alpha; \beta))$ and each $0 \leq i \leq c$. Hence we can decide whether $(q, \alpha) \models_{\mathsf{M}_c(\mathcal{D})} \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim i} \Phi_2)$ also for all $0 \leq i < c$. We can use these results to compute the satisfaction sets $\mathsf{Sat}[q, \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)]$ for each state $q \in Q$.

One approach would be, for each point $\alpha < \gamma < \beta$, and for each state $(q, [\alpha; \beta))$, to iterate over the individual values of γ; however, the size of intervals $[\alpha; \beta)$ in $\mathsf{Int}(q)$ for a given state q are dependent on the size of constants appearing in the time intervals ρ of the transitions $(q, \rho, \_) \in D$. We instead iterate over the size of the subscript c used in the temporal logic formula. More precisely, for each state $(q, [\alpha; \beta))$ of M_I, we have two cases.

$(q, [\alpha; \beta))$ **has a time-successor state.** (I.e. there exists a state $(q, [\beta; \beta')) \in Q_I$.) Then deciding whether $\gamma \in \mathsf{Sat}[q, \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)]$ for each $\alpha < \gamma < \beta$ can depend both on whether $\mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$ is satisfied in (q, α) *and* on the satisfaction of $\mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim i} \Phi_2)$ (for some i) in (q, β). For each $1 \leq j \leq \min(c, \beta - \alpha)$, we let $\beta - j \in \mathsf{Sat}[q, \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)]$ if and only if $((q, \alpha) \models_{\mathsf{M}_c(\mathcal{D})} \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)) \vee ((q, \beta) \models_{\mathsf{M}_c(\mathcal{D})} \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c-j} \Phi_2))$. Intuitively, the second conjunct corresponds to letting time pass and eventually moving to (q, β): if the formula with a subscript $c - j$ is satisfied j time units in the future, then the analogous formula with subscript c will be satisfied now. The first conjunct corresponds to taking an action transition: from the homogeneity of intervals with respect to action transitions, such a transition is available throughout the interval.

If $\beta - \alpha > c$, then for each $\alpha < j < \beta - c$ we let $j \in \mathsf{Sat}[q, \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)]$ if and only if $(q, \alpha) \models_{\mathsf{M}_c(\mathcal{D})} \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$.

$(q, [\alpha; \beta))$ **does not have a time-successor state.** In this case, for each $\alpha < j < \beta$, we let $j \in \mathsf{Sat}[q, \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)]$ if and only if $(q, \alpha) \models_{\mathsf{M}_c(\mathcal{D})} \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$.

We then merge adjacent intervals in $\mathsf{Sat}[q, \mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)]$. Analogously to the non-probabilistic case [17], the size of this set is bounded by $|\mathsf{Sat}[q, \Phi_1]| + |\mathsf{Sat}[q, \Phi_2]| + |D^q|$, and one can show that $|\mathsf{Sat}[q, \Psi]| \leq |\Psi| \cdot |D^q|$ for any PTCTL$[\leq, \geq]$ formula Ψ.

Observe that $|Q_I| \leq \sum_{q \in Q} |\mathsf{Int}(q)| \leq |\mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)| \cdot |D|$, and $| \rightarrow_I | \leq |Q_I| \cdot (1 + |D|)$. Recalling that the algorithm of Section 3.1 runs in time $O(c \cdot |Q_I| \cdot | \rightarrow_I |)$ when $\sim$ is $\leq$, we conclude that properties of the form $\mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\leq c} \Phi_2)$ can be verified in time $O(c \cdot |\mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\leq c} \Phi_2)|^2 \cdot |D|^3)$. Similarly, when $\sim$ is $\geq$, the corresponding algorithm of Section 3.1 runs in time $O((c \cdot |Q_I| \cdot | \rightarrow_I |) + poly(|\mathsf{M}_I|))$. The size of the TMDP M_I is no greater than $|Q_I| \cdot 2 \cdot |\mathcal{D}|$, and hence is no greater than $|\mathbb{P}_{\leq\lambda}(\Phi_1 \mathsf{U}_{\geq c} \Phi_2)| \cdot |D| \cdot 2 \cdot |\mathcal{D}|$. Hence, the algorithm when $\sim$ is $\geq$ runs in time $O((c \cdot |\mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\geq c} \Phi_2)|^2 \cdot |D|^3) + poly(|\mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\geq c} \Phi_2)| \cdot |D| \cdot |\mathcal{D}|))$.

These arguments can also be adapted for formulae $\mathbb{D}_{\bowtie\zeta}(\Phi')$. For a state s of a TMDP with a set of adversaries Adv, let $e^+_s(\Phi') = \sup_{A \in Adv} ExpectedTime^A_s(\Phi')$ and let $e^-_s(\Phi') = \inf_{A \in Adv} ExpectedTime^A_s(\Phi')$. In analogy with the case of properties of the form $\mathbb{P}_{\bowtie\lambda}(\Phi_1 \mathsf{U}_{\sim c} \Phi_2)$, for each state $(q, [\alpha; \beta)) \in Q_I$, we have $e^+_{(q,[\alpha;\beta))}(\Phi') = e^+_{(q,\alpha)}(\Phi')$ and $e^-_{(q,[\alpha;\beta))}(\Phi') = e^-_{(q,\alpha)}(\Phi')$. We apply the algorithm of de Alfaro [9] to M_I to compute $e^+_{(q,\alpha)}(\Phi')$ in the case of $\mathbb{D}_{\leq\zeta}(\Phi')$ and $e^-_{(q,\alpha)}(\Phi')$ in the case of $\mathbb{D}_{\geq\zeta}(\Phi')$.

To determine the values $e^+_{(q,\gamma)}(\Phi')$ and $e^-_{(q,\gamma)}(\Phi')$ for each $\alpha < \gamma < \beta$, we have two cases as above. If $(q, [\alpha; \beta))$ has a time-successor state, then for each $1 \leq j \leq \min(c, \beta - \alpha)$, we let $e^+_{(q,\beta-j)}(\Phi') = \max(e^+_{(q,\alpha)}(\Phi'), e^+_{(q,\beta)}(\Phi') + j)$, and similarly $e^-_{(q,\beta-j)}(\Phi') = \min(e^-_{(q,\alpha)}(\Phi'), e^-_{(q,\beta)}(\Phi') + j)$. If $\beta - \alpha > c$, then for each $\alpha < j < \beta - c$ we let $e^+_{(q,j)}(\Phi') = e^+_{(q,\alpha)}(\Phi')$ and $e^-_{(q,j)}(\Phi') = e^-_{(q,\alpha)}(\Phi')$.

On the other hand, if $(q, [\alpha; \beta))$ does not have a time-successor state, then for each $\alpha < j < \beta$, we let $e^+_{(q,j)}(\Phi') = e^+_{(q,\alpha)}(\Phi')$ and $e^-_{(q,j)}(\Phi') = e^-_{(q,\alpha)}(\Phi')$.

Then we can compare the obtained values of e^+ and e^- to the threshold ζ to decide whether $j \in \mathsf{Sat}[q, \mathbb{D}_{\bowtie\zeta}(\Phi')]$. We merge adjacent intervals in $\mathsf{Sat}[q, \mathbb{D}_{\bowtie\zeta}(\Phi')]$ to obtain the final satisfaction sets; as in the non-probabilistic case [17], the size of this set is bounded by $|D^q| + |\mathsf{Sat}[q, \Phi']| + 1$.

Verification of the $\mathbb{D}_{\bowtie\zeta}(\Phi')$ operator can be done in polynomial time in the size of M_I, and therefore our procedure takes time $poly(|\mathbb{D}_{\bowtie\zeta}(\Phi')| \cdot |D| \cdot |\mathcal{D}|)$).

Overall complexity. We obtain an overall time complexity of $O((|\Phi|^3 \cdot |D|^3 \cdot c_{max}) + poly(|\Phi| \cdot |D| \cdot |\mathcal{D}|))$. □

These two propositions imply that the program complexity of model checking PTCTL$[\leq, \geq]$ for the jump and continuous semantics is in P. This contrasts with the case of timed automata (with or without probability), where algorithms are based on the region graph and are exponential in the size of the system.

4 Complexity of Model Checking Durational Probabilistic Systems

In this section we consider upper and lower bounds on the complexity of model checking strongly non-Zeno DPSs. In particular we aim at comparing these results with those obtained for (non-probabilistic) durational systems, namely durational transition graphs (DTG) [17]. A DTG consists of a state set S, initial state s_{init}, and a labelling function l; in contrast to a DPS, however, the transition relation is of the form $\rightarrow \subseteq S \times \mathcal{I} \times S$. We know that model checking TCTL over DTGs is Δ^p_2-complete (resp. PSPACE-complete) with the jump semantics (resp. continuous semantics). Furthermore, model checking TCTL$[\leq, \geq]$ can be done in polynomial time for both semantics. We now identify cases in which the addition of probability makes model checking harder than in the non-probabilistic case, even for restricted sub-logics of PTCTL.

Complexity with probabilities 0/1. First we consider PTCTL$^{0/1}$, the "qualitative" sub-logic of PTCTL in which we allow $\mathbb{P}_{\bowtie\lambda}$ operators with $\lambda \in \{0, 1\}$ only, and in which the $\mathbb{D}_{\bowtie\zeta}$ operator is excluded.

Theorem 1 (Durational fully probabilistic systems). *Model checking* PTCTL$^{0/1}$ *over a strongly non-Zeno durational fully probabilistic system is a Δ^p_2-complete (resp. PSPACE-complete) problem for the jump (resp. continuous) semantics.*

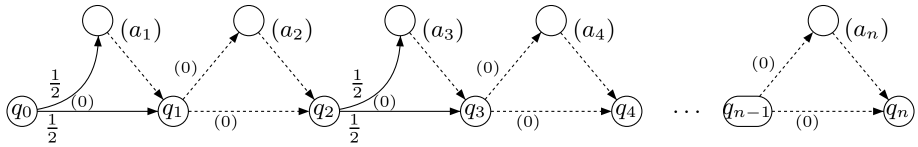

Fig. 2. The durational probabilistic system $\mathcal{D}_I$

Proof. This result derives mainly from the complexity of model checking over DTGs. Indeed, the general idea is to reduce model checking of $\text{PTCTL}^{0/1}$ over a strongly non-Zeno DFPS $\mathcal{D} = (Q, q_{init}, D, L)$ to TCTL model checking over the DTG $(S, s_{init}, \rightarrow, l)$ defined as follows: $S = Q$, $s_{init} = q_{init}$, $l = L$ and $(s, \rho, s') \in\rightarrow$ iff we have $(s, \rho, \mu) \in D$ and $\mu(s') > 0$. We replace $\text{PTCTL}^{0/1}$ subformulae by TCTL counterparts in the following way: $\mathbb{P}_{>0}(\varphi)$ is replaced by $\mathsf{E}\varphi$, while $\mathbb{P}_{\geq 1}(\mathsf{X}\Phi)$ (resp. $\mathbb{P}_{\geq 1}(\Phi_1 \mathsf{U}_{\leq c} \Phi_2)$, $\mathbb{P}_{\geq 1}(\Phi_1 \mathsf{U}_{=c} \Phi_2)$) is replaced by $\mathsf{AX}\Phi$ (resp. $\mathsf{A}(\Phi_1 \mathsf{U}_{\leq c} \Phi_2)$, $\mathsf{A}(\Phi_1 \mathsf{U}_{=c} \Phi_2)$). Finally, $\mathbb{P}_{\geq 1}(\Phi_1 \mathsf{U}_{\geq c} \Phi_2)$ is replaced by $\mathsf{A}(\Phi_1 \mathsf{U}_{\geq c} P_{\Phi_1 \mathsf{U} \Phi_2})$, where $P_{\Phi_1 \mathsf{U} \Phi_2}$ is a new atomic proposition that holds for states satisfying $\mathbb{P}_{\geq 1}(\Phi_1 \mathsf{U} \Phi_2)$. The standard PCTL model-checking algorithm [5], which runs in polynomial time, can be used to label states by $P_{\Phi_1 \mathsf{U} \Phi_2}$. Note that these reductions are possible because the DFPS is strongly non-Zeno. For the remaining $\text{PTCTL}^{0/1}$ formulae, as we are considering fully probabilistic systems, we have $\mathbb{P}_{<1}(\varphi) \equiv \neg\mathbb{P}_{\geq 1}(\varphi)$ and $\mathbb{P}_{\leq 0}(\varphi) \equiv \neg\mathbb{P}_{>0}(\varphi)$. The overall transformation provides Δ_2^p-membership (resp. PSPACE-membership) for the PTCTL model checking over DPS in the jump semantics (resp. continuous semantics).

With regard to the hardness results, we adapt the proofs used for DTGs with the same transformation of formulae as described above. □

Note that, following the results of [17] and using the translations of the proof of Theorem 1, we can find a polynomial-time algorithm for model checking DFPSs against formulae of $\text{PTCTL}^{0/1}$ without subscripts $=c$ in until modalities, both for the jump and continuous semantics.

Next, we address model checking of general, nondeterministic DPSs.

Theorem 2 (Durational probabilistic systems). *Model checking strongly non-Zeno durational probabilistic systems with the jump semantics is (1) PSPACE-hard for* $\text{PTCTL}^{0/1}$*, and (2) in PSPACE for* $\text{PTCTL}^{0/1}[\leq, \geq]$.

Proof. (1) We reduce a quantified version of the subset-sum problem, called *Q-subset-sum*, to a $\text{PTCTL}^{0/1}$ model-checking problem on strongly non-Zeno DPSs. As QBF can be reduced to Q-subset-sum, this suffices to show PSPACE-hardness. An instance I of Q-subset-sum contains a finite sequence X of integers $x_1, \ldots, x_n$, an integer G and a sequence of quantifiers $\mathcal{Q}_1, \ldots, \mathcal{Q}_n$ in $\{\exists, \forall\}$. The instance I is positive iff there exists a set Z of subsets of X s.t. (I) $\Sigma_{x \in X'} x = G$ for any $X' \in Z$ and (II) for any $Y \in Z$, if $\mathcal{Q}_i = \forall$, then there exists $Y' \in Z$ s.t. $x_j \in Y \Leftrightarrow x_j \in Y'$ for any $j < i$ and $x_i \in Y' \Leftrightarrow x_i \notin Y$. Assume w.l.o.g. that n is even and $\mathcal{Q}_{2i+1} = \forall, \mathcal{Q}_{2i+2} = \exists$ for all $0 \leq i < \frac{n}{2}$. Then we consider the DPS $\mathcal{D}_I$ described in Figure 2. The dashed lines correspond to non-deterministic choices, and the numbers in parentheses correspond to the duration of the transitions which they label.

Now assume $q_0 \models \neg\mathbb{P}_{<1}(\mathsf{F}_{=G} P)$ (where $\mathsf{F}_{\sim c}\text{-} \equiv \mathtt{true}\mathsf{U}_{\sim c}\text{-}$, and where q_n is the only state labelled with P): that is, there exists an adversary such that the probability of satisfying $\mathsf{F}_{=G}P$ from q_0 is 1. In terms of I, for any existential quantifier in I, it is possible to make a decision leading to a subset with exactly the sum G. Then $q_0 \models \neg\mathbb{P}_{<1}(\mathsf{F}_{=G}P)$ if and only if the instance I is positive.

(2) The PSPACE membership is shown as follows. For reasons of space we consider only the case $\mathbb{P}_{>0}(\Phi_1 \mathsf{U}_{\leq c} \Phi_2)$. Because the DPS is strongly non-Zeno, it suffices to verify that for any adversary there exists a path satisfying $\Phi_1 \mathsf{U}_{\leq c} \Phi_2$. We use the following algorithm which runs in polynomial space.

First note that $q \models \mathbb{P}_{>0}(\Phi_1 \mathsf{U}_{\leq d} \Phi_2)$ entails $q \models \mathbb{P}_{>0}(\Phi_1 \mathsf{U}_{\leq d+1} \Phi_2)$. For every state q we will compute the minimal d s.t. $\mathbb{P}_{>0}(\Phi_1 \mathsf{U}_{\leq d} \Phi_2)$ holds for q. First we define $T[q]$ as 0 (resp. ∞) if $q \models \Phi_2$ (resp. $q \not\models \Phi_1$). Then, for any $j = 0, 1, \ldots, c$, we try to update $T[q]$ for $q = q_1, \ldots, q_n$ if $T[q]$ has not yet been defined (where we enumerate the states in the topological order $\succ_0$). Updating $T[q]$ to j is done if, for any $(q, \rho, \mu) \in D$, there exists at least one state q' s.t. $\mu(q') > 0$ and $T[q'] \geq j - d_\rho$ where d_ρ is the maximal duration in ρ. Finally it remains to label a state q by $\mathbb{P}_{>0}(\Phi_1 \mathsf{U}_{\leq c} \Phi_2)$ iff $T[q] \leq c$. A similar procedure can be used to verify the other properties. □

For the continuous semantics, it is clear that model checking PTCTL is PSPACE-hard. These results show that strongly non-Zeno DFPSs are not harder to verify against PTCTL$^{0/1}$ than non-probabilistic durational systems against TCTL, and that combining probabilities and non-determinism induces a complexity blow-up for the jump semantics compared to the non-probabilistic case.

Complexity of full PTCTL. If we move from the sub-logic PTCTL$^{0/1}$ to the logic in which the operator $\mathbb{P}_{\bowtie\lambda}$ is permitted to have rational $\lambda \in [0; 1]$, we observe a complexity blow-up. It is sufficient to consider the simple formula $\mathbb{P}_{\geq\lambda}(\mathsf{F}_{\leq c}P)$ in the fully probabilistic case with the jump semantics.

Proposition 4. *Model checking $\mathbb{P}_{\geq\lambda}(\mathsf{F}_{\leq c}P)$ over durational fully probabilistic systems with the jump semantics is NP-hard.*

Proof (sketch). The proof consists in reducing the K-th largest subset problem, which is NP-hard [12, p. 225], to the problem of model checking a formula of the form $\mathbb{P}_{\geq\lambda}(\mathsf{F}_{\leq c}P)$ on a DFPS with the jump semantics. An instance I of K-th largest subset problem is a finite set $X = \{x_1, \ldots, x_n\}$ of natural numbers and two integers K and B. The problem consists in asking whether there are at least K distinct subsets $X' \subseteq X$ s.t. $\sum_{x \in X'} x \leq B$. Consider an adaptation of the DPS of Figure 2 where we replace the non-deterministic choices in states q_{2i+1}, for $0 \leq i < \frac{n}{2}$, by distributions with probabilities $\frac{1}{2}$, and recall that q_n is the only state labelled with P. This provides a DFPS that satisfies $\mathbb{P}_{\geq \frac{K}{2^n}}(\mathsf{F}_{\leq B}P)$ if and only if I is a positive instance. □

A corollary is that model checking PTCTL$[\leq, \geq]$ is NP-hard and coNP-hard over durational fully probabilistic systems with the jump semantics. Note that this problem is the simplest problem within our framework referring to quantitative temporal properties. It entails that considering simple timing constraints and quantitative probabilistic properties in the same model checking problem leads to NP-hardness, whereas considering *either* simple timing constraints (as in [17]) *or* quantitative probabilistic properties (as in [5]) allows for efficient model checking.

Table 1. Complexity results for model checking durational probabilistic systems

<table>
<tr><th></th><th colspan="2">Fully prob. DPS</th><th colspan="2">DPS</th></tr>
<tr><th></th><th>jump sem.</th><th>cont. sem.</th><th>jump sem.</th><th>cont. sem.</th></tr>
<tr><td>PTCTL$^{0/1}[\leq, \geq]$</td><td>P-complete</td><td>P-complete</td><td>P-hard
in PSPACE</td><td>P-hard
in EXPTIME$^{(\dagger)}$</td></tr>
<tr><td>PTCTL$^{0/1}$</td><td>Δ_2^p-complete</td><td>PSPACE-complete</td><td>PSPACE-hard
in EXPTIME</td><td>PSPACE-hard
in EXPTIME</td></tr>
<tr><td>PTCTL$[\leq, \geq]$</td><td colspan="4">NP-hard and coNP-hard
in EXPTIME$^{(\dagger)}$</td></tr>
<tr><td>PTCTL</td><td>Δ_2^p-hard
in EXPTIME</td><td>PSPACE-hard
in EXPTIME</td><td>PSPACE-hard
in EXPTIME</td><td>PSPACE-hard
in EXPTIME</td></tr>
</table>

For the general case where we have non-determinism, probabilities and PTCTL formulae, we conjecture that model checking is EXPTIME-complete. From the algorithms of Section 3 and the complexity results for PTCTL$^{0/1}$, we obtain the following corollary. Note that the EXPTIME-membership comes from a direct application of the algorithm described in Proposition 1 to $\mathsf{M}_j(\mathcal{D})$ or $\mathsf{M}_c(\mathcal{D})$.

Corollary 1. *Model checking* PTCTL *over durational probabilistic systems in the jump or continuous semantics is PSPACE-hard and it can be done in EXPTIME.*

5 Conclusion

In this paper we introduced durational probabilistic systems, a model to describe probabilistic, non-deterministic and timed systems. We showed how model checking can be done over this model, paying attention to complexity issues. Table 1 summarizes the results we presented in the paper. First, note that model checking can be done efficiently for fully probabilistic systems and qualitative PTCTL$^{0/1}$ properties without the exact time-bound subscript $= c$. However, as in the non-probabilistic case, adding the exact time-bound induces a complexity blow-up. This motivates the use of PTCTL$[\leq, \geq]$ where the subscripts in until formulae are restricted to $\leq c$ and $\geq c$ constraints. For this logic, even with quantitative properties, we have model checking algorithms running in time polynomial in $|\Phi| \cdot |\mathcal{D}|$ and linear in c_{max}, the maximal timing constant of the formula, as described in Proposition 2 and Proposition 3, and indicated by the $(\dagger)$ superscripts in the table. The precise polynomial depends on the kind of DPS and the choice of semantics. The formula's time constants are encoded in binary, and hence these algorithms belong to EXPTIME; nevertheless the algorithms should be interesting in practice, because they are polynomial in $|\mathcal{D}|$. In future work, we will consider the precise complexity of the non-complete model-checking problems listed in the table.

References

1. R. Alur, C. Courcoubetis, and D. L. Dill. Model-checking in dense real-time. *Information and Computation*, 104(1):2–34, 1993.

2. S. Andova, H. Hermanns, and J.-P. Katoen. Discrete-time rewards model-checked. In *Proc. 1st Int. Workshop on Formal Modeling and Analysis of Timed Systems (FORMATS 2003)*, volume 2791 of *LNCS*, pages 88–104. Springer, 2004.
3. C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, 2003.
4. C. Baier and M. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11(3):125–155, 1998.
5. A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Proc. 15th Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'95)*, volume 1026 of *LNCS*, pages 499–513. Springer, 1995.
6. S. Campos, E. M. Clarke, W. R. Marrero, M. Minea, and H. Hiraishi. Computing quantitative characteristic of finite-state real-time systems. In *Proc. IEEE Real-Time Systems Symposium (RTSS'94)*, pages 266–270. IEEE Computer Society Press, 1994.
7. E. M. Clarke, O. Grumberg, and D. Peled. *Model checking*. MIT Press, 1999.
8. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.
9. L. de Alfaro. *Formal verification of probabilistic systems*. PhD thesis, Stanford University, Department of Computer Science, 1997.
10. L. de Alfaro. Temporal logics for the specification of performance and reliability. In *Proc. 14th Annual Symp. on Theoretical Aspects of Computer Science (STACS'97)*, volume 1200 of *LNCS*, pages 165–176. Springer, 1997.
11. E. A. Emerson, A. K. Mok, A. P. Sistla, and J. Srinivasan. Quantitative temporal reasoning. *Real Time Systems*, 4(4):331–352, 1992.
12. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.
13. H. A. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
14. M. Kwiatkowska. Model checking for probability and time: From theory to practice. In *Proc. 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 351–360. IEEE Computer Society Press, 2003.
15. M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theoretical Computer Science*, 286:101–150, 2002.
16. F. Laroussinie, N. Markey, and P. Schnoebelen. On model checking durational Kripke structures (extended abstract). In *Proc. 5th Int. Conf. Foundations of Software Science and Computation Structures (FOSSACS 2002)*, volume 2303 of *LNCS*, pages 264–279. Springer, 2002.
17. F. Laroussinie, N. Markey, and P. Schnoebelen. Efficient timed model checking for discrete time systems. Submitted, 2004.
18. S. Tripakis. Verifying progress in timed systems. In *Proc. 5th AMAST Workshop on Real-Time and Probabilistic Systems (ARTS'99)*, volume 1601 of *LNCS*, pages 299–314. Springer, 1999.
19. M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. 16th Annual Symp. on Foundations of Computer Science (FOCS'85)*, pages 327–338. IEEE Computer Society Press, 1985.