Verification of Linear Duration Properties over Continuous Time Markov Chains*

Taolue Chen
Department of Computer Science
University of Oxford, UK

Marco Diciolla
Department of Computer Science
University of Oxford, UK

Marta Kwiatkowska
Department of Computer Science
University of Oxford, UK

Alexandru Mereacre
Department of Computer Science
University of Oxford, UK

ABSTRACT

Stochastic modeling and algorithmic verification techniques have been proved useful in analyzing and detecting unusual trends in performance and energy usage of systems such as power management controllers and wireless sensor devices. Many important properties are dependent on the cumulated time that the device spends in certain states, possibly intermittently. We study the problem of verifying *continuous-time Markov chains* (CTMCs) against *linear duration properties* (LDP), i.e. properties stated as conjunctions of linear constraints over the total duration of time spent in states that satisfy a given property. We identify two classes of LDP properties, eventuality duration properties (EDP) and invariance duration properties (IDP), respectively referring to the reachability of a set of goal states, within a time bound; and the continuous satisfaction of a duration property over an execution path. The central question that we address is how to compute the probability of the set of infinite timed paths of the CTMC that satisfy a given LDP. We present algorithms to approximate these probabilities up to a given precision, stating their complexity and error bounds. The algorithms mainly employ an adaptation of uniformization and the computation of volumes of multi-dimensional integrals under systems of linear constraints, together with different mechanisms to bound the errors.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Markov processes

General Terms

Verification

*This work is supported by the ERC Advanced Grant VERIWARE.

HSCC'12, April 17–19, 2012, Beijing, China.

Keywords

continuous-time Markov chains, linear duration logic, verification

1. INTRODUCTION

Stochastic modeling and verification [23] have become established as a means to analyze properties of system execution paths, for example dependability, performance and energy usage. Tools such as the probabilistic model checker PRISM [24] have been applied to model and verify many systems, ranging from embedded controllers and nanotechnology designs to wireless sensor devices and cloud computing, in some cases identifying flaws or unusual quantitative trends in system performance. The verification proceeds by subjecting a system model to algorithmic analysis against properties, typically expressed in probabilistic temporal logic, such as the probability of the vehicle hitting an obstacle is less than 10^{-4}, or the probability of an alarm bell ringing within 10 seconds is at least 95%. Many important properties, however, are dependent on the cumulated time that the system spends in certain states, possibly intermittently. Such *duration* properties, following the terminology of Duration Calculus (DC) [33], have been studied in the context of timed automata [1, 6, 22], but are not currently supported by existing probabilistic model checking tools. They can express, e.g., that the probability of an alarm bell ringing whenever the button has been pressed, possibly intermittently, for at least 2 seconds in total is at least 95%.

In this paper, we consider *Continuous-Time Markov Chain* (CTMC) models and study algorithmic verification for *linear duration properties* (LDP), i.e. properties involving linear constraints over cumulated residence time in certain states. CTMCs are widely used for performance and dependability analysis. CTMCs allow the modelling of real-time passage in conjunction with stochastic evolution governed by exponential distributions. They can be thought of as state transition systems, in which the system resides in a state on average for $1/r$ time units, where r is the exit rate, and transitions between the states are determined by a discrete probability distribution. As a concrete example of a system and property studied here, consider the dynamic power management system (DPMS) from [30], analysed in [29] against properties such as average power consumption. The DPMS includes a queue of requests, which have an exponentially distributed inter-arrival time, a power management controller and a service provider. The power management controller issues commands to the service provider depending on the power management policy, which involves switching between different power-saving modes. Fig. 1 depicts a CTMC model of the service provider for a Fujitsu disk drive. It consists of four states: *Busy*, *Idle*, *Standby* and *Sleep*. In this paper we are interested in computing the probability of, for instance, that *in 10 hours, the energy spent in Standby state is less than the energy spent in the Sleep state and the energy spent in the Idle state is less than one third of the energy spent in the Busy state*. We remark that the restriction to exponential distributions is not critical, since one can approximate any distribution by phase-type distributions, resulting in series-parallel combinations of exponential distributions [27].

The focus of CTMC model checking has primarily been on algorithms for specifications expressed in stochastic temporal logics, including *branching-time* variants, such as CSL [3], as well as *linear-time* temporal logic (LTL), whose verification reduces to the same problem for *embedded* discrete-time Markov chains (DTMCs). Model checking *deterministic* TA properties can be achieved by a reduction to computing the reachability probability in a *piecewise-deterministic Markov process* (PDP, [13]), based on the product construction between the CTMC and the DTA [10, 11, 4]. In [8], *time-bounded* verification of properties expressed by MTL or general TAs, which allow *nondeterminism*, is formulated. Approximation algorithms are proposed, based on path exploration of the CTMC, constraints generation and reduction to volume computation. There, "time-bounded" refers to the fact that only timed paths over a time interval of fixed, bounded length are considered, e.g. the probability of an alarm bell ringing whenever the button has been pressed for at least 2 seconds continuously. However, as pointed out in [1], the expressiveness of (D)TA/MTL is limited and *cannot* express *duration-bounded* causality properties which constrain the accumulated satisfaction times of state predicates along an execution path, visited possibly intermittently.

Contributions. We consider *linear duration formulas* (LDF) expressed as finite conjunctions of linear constraints on the cumulated time spent in certain states of the CTMC, see Eq. (1) for the precise formulation. Since we work with CTMCs, we interpret these formulas over finite and infinite *timed* paths. We distinguish two classes of linear duration properties. The difference lies in how to interpret LDF over *infinite* timed paths.

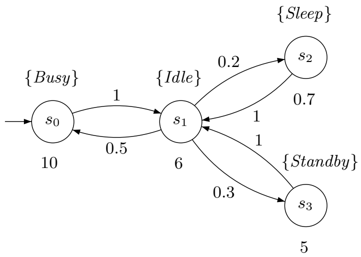
- *Eventuality Duration Property (EDP).* Similarly to [1, 22], given a set of goal states G, an infinite path is said to satisfy LDF if its prefix until G is reached satisfies EDP. We identify two variants, the timed-bounded case ($T < \infty$) and unbounded case ($T = \infty$).
- *Invariance Duration Property (IDP).* Similarly to [6], we require that *each* prefix of the infinite path satisfies LDF, again distinguishing the timed-bounded case ($T < \infty$) and the unbounded case ($T = \infty$). We remark that, in duration calculus, a stronger requirement is imposed, i.e., any fragment (not only the prefix, but also starting from an arbitrary state) of the infinite path must satisfy LDF. We do not adopt this view, as we work in the traditional setting of temporal logics, rather than an interval temporal logic.

The central questions we consider is how to compute the probability of the set of timed paths of the CTMC which satisfy linear-time properties expressed as LDF. To the best of our knowledge, this is the first paper that considers duration properties for CTMCs. We now give a brief account of the techniques introduced in this paper.

We propose two approaches to verify the timed-bounded variant of EDP. First, we define a system of partial differential equations (PDEs) and a system of integral equations whose solutions capture the probability that an EDP is satisfied on a given CTMC. Second, we leverage the uniformization method [21], which reduces the problem to computing the probability of a set of finite timed paths under a system of linear constraints. This can be solved through the computation of volumes of convex polytopes in the general case, while, in the case that the LDF only involves one conjunct, it can be reduced to the computation of order statistics, which is more efficient. In the unbounded case, by exploiting Markov inequality, we show how to approximate the probability by choosing a sufficiently large time-bound. This is of independent interest, and can be used to improve our previous results [11, 8]. To verify an IDP, in the unbounded case we perform a graph analysis of the CTMC according to the LDF, and thus obtain a variant of EDP, which can be solved by extending the approaches developed in the previous case. In the time-bounded case, transient analysis of the CTMC is needed.

We remark that linear duration properties are closely related to *Markovian Reward Models* (MRM, [2]), which are CTMCs augmented with multiple reward structures assigning real-valued rewards to each state in the model. Properties of MRMs can be expressed in continuous stochastic reward logic (CSRL, [2]). CSRL model checking for MRMs [17, 12] involves timed-bounded and/or reward-bounded reachability problems, which can be formulated in terms of model checking of LDP, over CTMCs, by treating the rewards in MRM as coefficients of linear duration formulas. (This will be made clearer in Sect. 2.3.) We emphasise that, in contrast to [12], as the coefficients in LDF might be negative, we can deal with CSRL in MRMs with arbitrary rewards. The link to MRM (with arbitrary rewards) is beneficial, as energy constraints [7] studied in TA can be naturally adapted to stochastic models (like CTMCs), and can be solved by approaches presented in the current paper.

Related Work. Algorithmic verification of duration properties has primarily been studied in the setting of TA, for instance [1, 6, 22]. Similarly to our setting, TA also admit the unfolding of the system into timed execution paths, except that we have to calculate the probability of the set of paths satisfying a given property, rather than quantifying over their existence. The "duration bounded reachability" problem of [1] can be viewed as a subclass of EDP, in view of the requirement that all coefficients appearing in the linear constraints are nonnegative. Reachability for *integral graphs* [22] can be reduced to verification of EDP for TA, which is solved by mixed linear-integer programming. [6] extended branching real-time logic TCTL with duration constraints and studied response/persistence properties. For DC, which is based on interval temporal logic that differs from our setting, the focus has been on so called *linear durational invariants* (LDI, [34]). Again, TA (and their subclasses or extensions) are considered, and different techniques are proposed, for instance, reduction to linear programming or CTL, dis-



$$\mathbf{Q} = \begin{bmatrix} -10 & 10 & 0 & 0 \\ 3 & -6 & 1.2 & 1.8 \\ 0 & 0.7 & -0.7 & 0 \\ 0 & 5 & 0 & -5 \end{bmatrix}$$

Figure 1: An example CTMC

cretization, etc. We mention, e.g., [26, 31, 32], which are specific to TA and cannot be adapted to CTMCs.

There is only scant work addressing probabilistic/stochastic extensions of DC. Simple Probabilistic Duration Calculus, interpreted over (finite-state) continuous semi-Markov processes, is introduced in [20], together with the associated axiomatic system, and applied to QoS contracts in [16]. However, algorithmic verification is not addressed. [19] studied verification problems of (subclasses of) LDI in the setting of probabilistic TA which only involves discrete probabilities. The technique is an adaption of discretization for TA.

We also mention [5], which considers CTL and LTL extended with prefix-accumulation assertions for a quantitative extension of Kripke structure. (Un)decidability results are obtained. The prefix-accumulation assertions are similar to our linear constraints modulo the difference between models under consideration (CTMCs are a continuous model with randomization, whereas Kripke structures are a discrete model without randomization.) For further discussion, we refer the reader to the full version of the paper [9].

2. PRELIMINARIES

2.1 Continuous-time Markov chains

Given a set $\mathcal{H}$, let Pr: $\mathcal{F}(\mathcal{H}) \to [0,1]$ be a *probability measure* on the measurable space $(\mathcal{H}, \mathcal{F}(\mathcal{H}))$, where $\mathcal{F}(\mathcal{H})$ is a σ-algebra over $\mathcal{H}$.

Definition 1 *[CTMC] A (labeled) continuous-time Markov chain* (CTMC) *is a tuple* $\mathcal{C} = (S, \mathrm{AP}, L, \alpha, \mathbf{P}, E)$ *where:* S *is a finite set of states;* AP *is a finite set of atomic propositions;* $L : S \to 2^{\mathrm{AP}}$ *is the labeling function;* α *is the initial distribution over* S*;* $\mathbf{P} : S \times S \to [0,1]$ *is a stochastic matrix; and* $E : S \to \mathbb{R}_{\geq 0}$ *is the exit rate function.*

Example 1 *An example* CTMC *is illustrated in Fig. 1, where* AP $= \{Busy, Idle, Sleep, Standby\}$ *and* $\alpha(s_0) = 1$ *is the initial distribution. The exit rates are indicated at the states, whereas the transition probabilities are attached to the transitions. The CTMC is a model of the service provider of the DPMS system described in Sect. 1.*

In a CTMC $\mathcal{C}$, state residence times are *exponentially* distributed. More precisely, the residence time of the state $s \in S$ is a random variable governed by an exponential distribution with parameter $E(s)$. Hence, the probability to exit state s in t time units (t.u. for short) is given by $\int_0^t E(s) \cdot e^{-E(s)\tau} d\tau$; and the probability to take the transition from s to s' in t t.u. equals $\mathbf{P}(s,s') \cdot \int_0^t E(s) \cdot e^{-E(s)\tau} d\tau$. A state s is *absorbing* if $\mathbf{P}(s,s') = 1$. We also define the *infinitesimal generator* $\mathbf{Q}$ of $\mathcal{C}$ as $\mathbf{Q} = \mathbf{E} \cdot \mathbf{P} - \mathbf{E}$, where $\mathbf{E}$ is the diagonal matrix with exit rates on diagonal. Occasionally we use $X(t)$ to denote the underlying *stochastic process* of $\mathcal{C}$. We write $\pi(t)$ for the *transient probability distribution*, where, for each $s \in S$, $\pi_s(t) = \Pr\{X(t) = s\}$ is the probability to be in state s at time t. It is well-known that $\pi(t)$ completely depends on the initial distribution α and the infinitesimal generator $\mathbf{Q}$, i.e., it is the solution of the Chapman-Komogorov equation $\frac{d\pi(t)}{dt} = \pi(t)\mathbf{Q}$ and $\pi(0) = \alpha$. Note that efficient algorithms (e.g. uniformization approach, cf. Sect. 3.1.2, Eq. (3)) exist to compute $\pi(t)$.

An *infinite timed* path in $\mathcal{C}$ is an infinite sequence $\rho = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \cdots \xrightarrow{t_{n-1}} s_n \ldots$; and a *finite timed* path is a finite sequence $\sigma = s_0 \xrightarrow{t_0} \cdots \xrightarrow{t_{n-1}} s_n$. In both cases we assume that $t_i \in \mathbb{R}_{>0}$ for each $i \geq 0$; moreover, we write $\rho[0..n]$ for σ. Below we usually follow the convention to let ρ (resp. σ) range over infinite (resp. finite) timed paths, unless otherwise stated. We define $|\sigma| := n$ to be the length of a finite timed path σ. For a finite or infinite path θ, $\theta[n] := s_n$ is the $(n+1)$-th state of θ and $\theta\langle n\rangle := t_n$ is the time spent in state s_n; let $\theta@t$ be the state occupied in θ at time $t \in \mathbb{R}_{\geq 0}$, i.e. $\theta@t := \theta[n]$, where n is the smallest index such that $\sum_{i=0}^{n} \theta\langle i\rangle \geq t$. Let $Paths^{\mathcal{C}}$ denote the set of infinite timed paths in $\mathcal{C}$, with abbreviation *Paths* when $\mathcal{C}$ is clear from the context. Intuitively, a timed path ρ suggests that the CTMC $\mathcal{C}$ starts in state s_0 and stays in this state for t_0 t.u., and then jumps to state s_1, staying there for t_1 t.u., and then jumps to s_2 and so on. An example timed path is $\rho = s_0 \xrightarrow{3} s_1 \xrightarrow{2} s_0 \xrightarrow{1.5} s_1 \xrightarrow{3.4} s_2 \ldots$ with $\rho[2] = s_0$ and $\rho@4 = \rho[1] = s_1$.

Sometimes we refer to *discrete time Markov chains* (DTMCs), denoted $\mathcal{D} = (S, \mathrm{AP}, \alpha, L, \mathbf{P})$, where the components of the tuple have the same meaning as those of CTMCs defined in Def. 1. In particular, we say such $\mathcal{D}$ is the *embedded DTMC* of the CTMC $\mathcal{C}$. Similarly, a (finite) *discrete* path $\varsigma = s_0 \to s_1 \to \ldots$ is a (finite) sequence of states; $\varsigma[n]$ denotes the state s_i, $\varsigma[0..n]$ denotes the prefix of length n of ς, and $|\varsigma|$ denotes the length of ς (in case that ς is finite). We also define $Paths^{\mathcal{D}}$ to be the set of all infinite paths of the DTMC $\mathcal{D}$. Given a finite *discrete* path $\varsigma = s_0 \to \cdots \to s_n$ of length n and $x_0, \ldots, x_{n-1} \in \mathbb{R}_{>0}$, we define $\varsigma[x_0, \ldots, x_{n-1}]$ to be the finite *timed* path σ such that $\sigma[i] := s_i$ and $\sigma\langle i\rangle := x_i$ for each $0 \leq i < n$. Let $\Gamma \subseteq \mathbb{R}^n_{>0}$, then $\varsigma[\Gamma] = \{\varsigma[x_0, \ldots, x_{n-1}] \mid (x_0, \ldots, x_{n-1}) \in \Gamma\}$.

The definition of a *Borel space* on timed paths of CTMCs follows [3]. A CTMC $\mathcal{C}$ yields a probability measure $\Pr^{\mathcal{C}}_{\alpha}$ on $Paths^{\mathcal{C}}$ as follows. Let $s_0, \ldots, s_k \in S$ with $\mathbf{P}(s_i, s_{i+1}) > 0$ for $0 \leq i < k$ and $I_0, \ldots, I_{k-1}$ be nonempty intervals in $\mathbb{R}_{\geq 0}$. Let $C(s_0, I_0, \ldots, I_{k-1}, s_k)$ denote the *cylinder set* consisting of all $\rho \in Paths$ such that $\rho[i] = s_i$ $(0 \leq i \leq k)$ and $\rho\langle i\rangle \in I_i$ $(0 \leq i < k)$. $\mathcal{F}(Paths)$ is the smallest σ-algebra on *Paths* which contains all sets $C(s_0, I_0, \ldots, I_{k-1}, s_k)$ for all state sequences $(s_0, \ldots, s_k) \in S^{k+1}$ with $\mathbf{P}(s_i, s_{i+1}) > 0$ for $(0 \leq i < k)$ and $I_0, \ldots, I_{k-1}$ ranging over all sequences of nonempty intervals in $\mathbb{R}_{\geq 0}$. The *probability measure* $\Pr^{\mathcal{C}}_{\alpha}$

on $\mathcal{F}(Paths)$ is the unique measure defined by induction on k by $\Pr^{\mathcal{C}}_{\alpha}(C(s_0)) = \alpha(s_0)$ and for $k > 0$:

$$\Pr^{\mathcal{C}}_{\alpha}(C(s_0, I_0, \ldots, I_{k-1}, s_k)) = \Pr^{\mathcal{C}}_{\alpha}(C(s_0, I_0, \ldots, I_{k-2}, s_{k-1})) \times \int_{I_{k-1}} \mathbf{P}(s_{k-1}, s_k) E(s_{k-1}) \cdot e^{-E(s_{k-1})\tau} d\tau.$$

Sometimes we write Pr instead of $\Pr^{\mathcal{C}}_{\alpha}$ when $\mathcal{C}$ and α are clear from the context. Elements of the σ-algebra denote events in the probability space. We now define two such events that will be needed later.

Definition 2 *Given a CTMC $\mathcal{C}$ and $B \subseteq S$, we define:*

- $\Diamond^{\leq T} B = \{\rho \in Paths^C \mid \exists n.\rho[n] \in B$ *and* $\sum_{i=0}^{n} \rho\langle i\rangle \leq T\}$, *i.e.,* $\Diamond^{\leq T} B$ *denotes the set of timed paths which reach B in time interval $[0, T]$. Note that $\Pr^{\mathcal{C}}(\Diamond^{\leq T} B)$ can be computed by a reduction to the computation of the transient probability distribution; see [3].*
- $\Diamond B = \{\rho \in Paths^C \mid \exists n.\rho[n] \in B\}$, *i.e., $\Diamond B$ denotes the set of timed paths which reach B. (It is the unbounded variant of $\Diamond^{\leq T} B$.) Note that $\Pr^{\mathcal{C}}(\Diamond B)$ is essentially the reachability probability of B in the embedded DTMC of $\mathcal{C}$; see [3]. Moreover, we write* $\mathrm{Prob}(s, \Diamond B)$ *for the reachability probability of B* when starting from the state s.

2.2 Duration Properties

We first introduce a language which includes the propositional calculus augmented with the *duration function* $\int$ and linear inequalities. In the remainder of this section, we assume a CTMC $\mathcal{C} = (S, \mathrm{AP}, L, \alpha, \mathbf{P}, E)$.

State formulas, defined in the usual way over the propositions in AP and the boolean operators, can be evaluated over single states of CTMCs using the interpretation assigned to them by the labeling function L (see Def. 1). The *duration function* $\int$ is interpreted over a *finite* timed path. Let ap be a state formula and $\sigma = s_0 \xrightarrow{t_0} \ldots \xrightarrow{t_{n-1}} s_n$. The value of $\int ap$ for σ, denoted $[\![ap]\!]_\sigma$, is defined as $\sum_{0 \leq i < n, \sigma[i] \models ap} t_i$. That is, the value of $\int ap$ equals the sum of durations spent in states satisfying ap.

A *linear duration formula* (LDF) is of the form

$$\varphi = \bigwedge_{j \in J} \left(\sum_{k \in K_j} c_{jk} \int ap_{jk} \leq M_j \right), \tag{1}$$

where $c_{jk}, M_j \in \mathbb{R}$, ap_{jk} are state formulas, and J, K_j for $j \in J$ are finite index sets. Below we usually assume that $J = \{0, \cdots, m\}$.

Remark 1 *We did not introduce the* disjunction *or (more general) boolean operators in Eq.* (1) *for simplicity. All our results can be generalized to these cases by the* exclusion-inclusion principle *[28], paying the price of higher complexity.*

Definition 3 *Given a finite timed path $\sigma = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots \xrightarrow{t_{n-1}} s_n$ and an LDF φ of the form defined in Eq.* (1), *we write $\sigma \models \varphi$ if for each $j \in J$, $\sum_{k \in K_j} c_{jk} \cdot [\![ap_{jk}]\!]_\sigma \leq M_j$.*

Example 2 *For the* CTMC *in Fig. 1, the LDF $\varphi = \int Idle - \frac{1}{3} \int Busy \leq 0$ expresses the constraint that during the evolution of the* CTMC *the accumulated time spent in the Idle state must be less than or equal to one third of the accumulated time spent in the Busy state.*

Inspired by the notation of [34], we shall also work on a slight extension of LDF, i.e., formulas of the form:[1]

$$\Phi := \int 1 \leq T \rightarrow \varphi,$$

where $T \in \mathbb{R}_{\geq 0} \cup \{\infty\}$. According to Def. 3, $\int 1$ denotes the total time spent on a finite timed path σ. Hence $\sigma \models \Phi$ if φ holds whenever the total time of σ is less or equal than T. Note that, if $T = \infty$, Φ simply degenerates to φ.

In general, given a CTMC and a duration property specified by an LDF, we are interested in computing the probability of *infinite timed* paths satisfying the LDF. We now generalize the satisfaction relation on finite paths, as defined in Def. 3, to *infinite* paths. Here we have two options, i.e., the *finitary* and *infinitary* conditions. The former is motivated by standard automata theory, while the latter is natural when one thinks of "globally" (e.g., the $\Box$ operator in LTL).

Definition 4 *Let $\rho = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots$ be an* infinite *timed path and φ (or Φ) be an LDF.*

1. *Finitary satisfaction condition. Given a set of* goal *states $G \subseteq S$, we write $\rho \models^G \varphi$ if there exists the* first *$i \in \mathbb{N}$ such that (1) $\rho[i] \in G$, and (2) $\rho[0..i] \models \varphi$ (cf. Def. 3). Furthermore, we write $\rho \models^G_T \varphi$ for a given $T \in \mathbb{R}_{\geq 0}$, and if, in addition to (1) and (2), $\sum_{j=0}^{i-1} \rho\langle j\rangle \leq T$ holds.*
2. *Infinitary satisfaction condition. We write $\rho \models^\star \varphi$ if for any $n \geq 0$, $\rho[0..n] \models \varphi$ (cf. Def. 3).*

Problem Statements. Corresponding to Def. 4, we focus on algorithmic verification problems for two classes of LDP, i.e., *Eventuality Duration Property* (EDP) and *Invariance Duration Property* (IDP), as follows.

- **Verification of EDP.** Formally, given a CTMC $\mathcal{C}$, a set of *goal* states $G \subseteq S$, and an LDF $\Phi = \int 1 \leq T \rightarrow \varphi$, compute the probability of the set of infinite timed paths of $\mathcal{C}$ satisfying Φ under the *finitary satisfaction condition*. Depending on T, we distinguish two cases:
  - Time-bounded case: $T < \infty$, for which we denote the desired probability by $\boxed{\mathrm{Prob}(\mathcal{C} \models^G \Phi)}$.
  - Unbounded case: $T = \infty$, for which we denote the desired probability by $\boxed{\mathrm{Prob}(\mathcal{C} \models^G \varphi)}$. Note that this is valid as, in this case, Φ is simply equivalent to φ.

  The algorithms for these two cases are given in Sect. 3.1 and Sect. 3.2, respectively.

[1] Note that 1 denotes "true", $\rightarrow$ denotes "imply" and $\int 1 \leq T \rightarrow \varphi$ is a single formula.

- **Verification of IDP.** Formally, given a CTMC $\mathcal{C}$ and an LDF $\Phi = \int 1 \leq T \to \varphi$, compute the probability of the set of infinite timed paths of $\mathcal{C}$ satisfying Φ under the *infinitary satisfaction condition*. We also have two cases, i.e., the time-bounded case and unbounded case, which we denote by $\boxed{\mathrm{Prob}(\mathcal{C} \models^{\star} \Phi)}$ and $\boxed{\mathrm{Prob}(\mathcal{C} \models^{\star} \varphi)}$, respectively. The algorithms for these two cases are given in Sect. 4.2 and Sect. 4.1, respectively.

2.3 Relationship to MRMs

Definition 5 *[MRM] A (labeled) MRM $\mathcal{M}$ is a pair $(\mathcal{C}, \mathbf{r})$ where $\mathcal{C}$ is CTMC, and $\mathbf{r} : S \to \mathbb{R}^d$ is a* reward structure *which assigns to each state $s \in S$ a vector of rewards $(r_1(s), \cdots, r_d(s))$.*

Remark 2 *The MRM defined in Def. 5 is more general than the one in [2], in the sense that we have* multiple reward structures, *and, more importantly, we allow arbitrary (instead of nonnegative) rewards associated with the states.*

For a CTMC $\mathcal{C}$ and LDF φ, we show how to construct an MRM $\mathcal{C}[\varphi]$. For every state $s_i \in S$, we define $r_{ji} = \sum_{t \in K_j, s_i \models ap_{jt}} c_{jt}$ for all $j \in J$. This yields a multiple reward structure $\mathbf{r}$ with $\mathbf{r}(s_i) = (r_{0i}, \cdots, r_{(|J|-1)i})$. Hence $\mathcal{C}[\varphi] = (\mathcal{C}, \mathbf{r})$. It is straightforward to see that the constraint expressed by LDF can be alternatively formulated as the "reward-bounded" constraint for MRMs, since $\sum_{k \in K_j} c_{jk} \int ap_{jk}$ essentially denotes the accumulated rewards along a finite timed path, and hence M_j can be regarded as the bound of the reward.

On the other hand, given an MRM and a vector of reward bounds M_j for each reward structure, we construct an LDF φ as $\bigwedge_{j \in J} \sum_{s \in S} r_j(s) \int @s \leq M_j$, where $@s$ is an atomic proposition which holds exactly at state s. Hence, the reward-bounded verification problem for MRMs can be encoded into verification of linear duration properties in CTMCs.

It is easy to see that this correspondence, stated in the unbounded case, can be adapted to the time-bounded case without any difficulties.

3. VERIFICATION OF EDP

Throughout this section, we fix a CTMC $\mathcal{C} = (S, \mathrm{AP}, L, \alpha, \mathbf{P}, E)$ and an LDF $\Phi = \int 1 \leq T \to \bigwedge_{j \in J} (\sum_{k \in K_j} c_{jk} \int ap_{jk} \leq M_j)$.

3.1 Time-bounded Verification of EDP

Our task is to compute $\mathrm{Prob}(\mathcal{C} \models^G \Phi)$. First observe that

Proposition 1 *Given a* CTMC $\mathcal{C}$ *and an LDF* Φ*, we have:*

$$\mathrm{Prob}(\mathcal{C} \models^G \Phi) = \Pr(\Diamond G) - \Pr(\Diamond^{\leq T} G) + \mathrm{Prob}(\mathcal{C} \models^G_T \varphi).$$

Recall that $\Pr(\Diamond G)$ and $\Pr(\Diamond^{\leq T} G)$ can be easily computed (cf. Def. 2). Hence, the remaining of this section is devoted to computing $\mathrm{Prob}(\mathcal{C} \models^G_T \varphi) := \Pr(\{\rho \mid \rho \models^G_T \varphi\})$, i.e. the probability of the set of paths of the CTMC $\mathcal{C}$, which reach G in time interval $[0, T]$ and satisfy the LDF φ before that happens; see Def. 4(1).

3.1.1 PDE and Integral Formulations

In order to compute $\mathrm{Prob}(\mathcal{C} \models^G_T \varphi)$, we shall use the link to MRMs established in Sect. 2.3. Recall that $\mathcal{C}[\varphi]$ is the MRM obtained from $\mathcal{C}$ and φ. We need an extra transformation over $\mathcal{C}[\varphi]$, namely, making each state $s \in G$ absorbing, and set $\mathbf{r}(s) = (0, \cdots, 0)$ (i.e., the rewards associated with s are all 0). We denote the resulting MRM by $\mathcal{C}[\varphi, G]$. Recall that $X(t)$ is the underlying stochastic process of the CTMC $\mathcal{C}$. We denote by $\mathbf{Y}(T)$ the vector of accumulated rewards in the MRM $\mathcal{C}[\varphi]$ (see Sect. 2.3) up to time T, i.e. $\mathbf{Y}(T) = (Y_0(T), \ldots, Y_{|J|-1}(T))$ and each $Y_j(T)$ ($j \in J$) corresponds to a reward structure in CTMC $\mathcal{C}$. The vector of stochastic processes $\mathbf{Y}(T)$ is fully determined by $X(T)$ and the vector of reward structures of the state s is $\mathbf{r}(s_i) = (r_{0i}, \ldots, r_{(|J|-1)i})$, because $\mathbf{Y}(t) = \int_0^t \mathbf{r}(X(\tau)) d\tau$.

Define $\mathbf{F}(T, \mathbf{y})$ to be the matrix of the joint probability distribution of state and rewards with entries $\mathbf{F}(T, \mathbf{y})[s, s'] = F_s^{s'}(T, \mathbf{y})$ for $s, s' \in S$ and

$$F_s^{s'}(T, \mathbf{y}) = \Pr\left\{ X(T) = s', \bigwedge_{j \in J} Y_j(T) \leq y_j \mid X(0) = s \right\},$$

where $\mathbf{y} = (y_0, \cdots, y_{|J|-1})$. Note that, we define $\mathbf{F}(T, \mathbf{y})$ over the induced MRM $\mathcal{C}[\varphi, G]$.

Theorem 1 *Given a* CTMC $\mathcal{C}$*, an LDP formula φ, a vector $\mathbf{M} = (M_0, \ldots, M_{|J|-1})$, where M_j's are defined as in φ (cf. Eq. (1)) and a set of goal states G, we obtain the induced* MRM $\mathcal{C}[\varphi, G]$*, and we have:*

$$\mathrm{Prob}(\mathcal{C} \models^G_T \varphi) = \sum_{s \in S} \sum_{s' \in G} \alpha(s) F_s^{s'}(T, \mathbf{M}).$$

Thm. 1 suggests a reduction to $\mathbf{F}(t, \mathbf{y})$, which we now characterize in terms of a system of PDEs.

Theorem 2 *For an* MRM $\mathcal{C}[\varphi, G]$*, the function $\mathbf{F}(t, \mathbf{y})$ is given by the following system of PDEs:*

$$\frac{\partial \mathbf{F}(t, \mathbf{y})}{\partial t} + \sum_{j \in J} \mathbf{D}_j \cdot \frac{\partial \mathbf{F}(t, \mathbf{y})}{\partial y_j} = \mathbf{Q} \cdot \mathbf{F}(t, \mathbf{y}), \qquad (2)$$

where $\mathbf{D}_j$ is a diagonal matrix such that $\mathbf{D}_j(s, s) = r_j(s)$.

The system of PDEs from Theorem 2 is a special case of the system of PDEs derived from Petri net specifications [18] and PDP models [13].

Example 3 *For the* CTMC *depicted in Fig.1, with $r(s_0) = 1$ and $r(s_1) = -1$, we can derive the following system of PDEs:*

$$\begin{aligned}
\frac{\partial F_{s_0}^{s_1}(t, y)}{\partial t} + \frac{\partial F_{s_0}^{s_1}(t, y)}{\partial y} &= 10 F_{s_1}^{s_1}(t, y) - 10 F_{s_0}^{s_1}(t, y), \\
\frac{\partial F_{s_1}^{s_0}(t, y)}{\partial t} - \frac{\partial F_{s_1}^{s_0}(t, y)}{\partial y} &= -6 F_{s_1}^{s_0}(t, y) + 3 F_{s_0}^{s_0}(t, y), \\
&\quad + 1.2 F_{s_2}^{s_0}(t, y) + 1.8 F_{s_3}^{s_0}(t, y).
\end{aligned}$$

We next provide an alternative characterization in terms of a system of integral equations, as follows.

Theorem 3 *The solution of the system of PDEs in Eq.* (2) *is the least fixpoint of the following system of integral equations:*

$$F_s^{s'}(t,\mathbf{y}) = e^{\mathbf{Q}(s,s)t}F_s^{s'}(0,\mathbf{y}-\mathbf{r}(s)t) + \int_0^t \sum_{z\neq s} e^{\mathbf{Q}(s,s)x}\mathbf{Q}(s,z)F_z^{s'}(t-x,\mathbf{y}-\mathbf{r}(s)x)dx.$$

Thm. 2 and Thm. 3 imply that, to solve the bounded-time EDP verification problem, we need to solve (first-order) PDEs or integral equations. However, this is usually costly and numerically unstable [15]. We present solutions in the next section, based on uniformization.

3.1.2 Uniformization algorithm

In this section we present a uniformization algorithm to compute $F_s^{s'}(t,\mathbf{y})$. The *uniformization* method [21] consists in transforming the CTMC $\mathcal{C}$ into a behaviorally equivalent DTMC $\mathcal{D}$. (NB. this is *not* the embedded DTMC of $\mathcal{C}$.) The state space and initial distribution of $\mathcal{D}$ are the same as for $\mathcal{C}$. The probability matrix $\widehat{\mathbf{P}}$ of $\mathcal{D}$ is constructed by $\widehat{\mathbf{P}} = \mathbf{I} - \frac{\mathbf{Q}}{\Lambda}$, where Λ is the maximal exit rate of $\mathcal{C}$. We obtain

$$\pi(t) = e^{(\widehat{\mathbf{P}}-\mathbf{I})\Lambda t} = \sum_{n=0}^{\infty} \widehat{\mathbf{P}}^n \frac{(\Lambda t)^n}{n!} e^{-\Lambda t}. \tag{3}$$

We can apply the uniformization technique to efficiently compute $F_s^{s'}(t,\mathbf{y})$. First, we note that the infinite sum in Eq. (3) represents the probability $\frac{(\Lambda t)^n}{n!}e^{-\Lambda t}$ that exactly n Poisson arrivals occur in an interval of time $[0,t)$ multiplied with the probability $\widehat{\mathbf{P}}^n$ to take the state transitions corresponding to the arrivals. Then using Eq. (3) we obtain

$$F_s^{s'}(t,\mathbf{y}) = \sum_{n=0}^{\infty} e^{-\Lambda t}\frac{(\Lambda t)^n}{n!} \cdot \Bigg(\sum_{\substack{\varsigma\in Paths^{\mathcal{D}} \\ |\varsigma|=n}} \Pr\{\varsigma \mid X(0)=s\} \cdot \Pr\{X(n)=s', \mathbf{Y}(t)\leq \mathbf{y} \mid \varsigma\}\Bigg),$$

where, for a given path $\varsigma = s \to s_1 \to \cdots \to s_{n-1} \to s'$, $\Pr\{\varsigma \mid X(0)=s\} = \widehat{\mathbf{P}}(s,s_1)\times\cdots\times\widehat{\mathbf{P}}(s_{n-1},s')$ and $\Pr\{X(n)=s',\mathbf{Y}(t)\leq\mathbf{y}\mid\varsigma\}$ denotes the conditional probability that given the path ς at step n the state is s' and the total accumulated reward until time t is less than $\mathbf{y}$. The above equation can also be written as

$$F_s^{s'}(t,\mathbf{y}) = \sum_{n=0}^{\infty} e^{-\Lambda t}\frac{(\Lambda t)^n}{n!} \sum_{\substack{\varsigma\in Paths^{\mathcal{D}} \\ |\varsigma|=n \\ \varsigma[0]=s \\ \varsigma[n]=s'}} \mathrm{Prob}(\varsigma)\cdot\Pr\{\mathbf{Y}(t)\leq\mathbf{y}\mid\varsigma\}. \tag{4}$$

Now the task is to compute $\Pr\{\mathbf{Y}(t)\leq\mathbf{y}\mid\varsigma\}$. We first present a general approach based on linear constraints.

Approach based on linear constraints.

We can calculate $\Pr\{\mathbf{Y}(t)\leq\mathbf{y}\mid\varsigma\}$ by reducing it to the computation of the volume of a convex polytope. The basic idea is to generate timed constraints over variables determining the residence time of each state along ς to make $\mathbf{Y}(t)\leq\mathbf{y}$ hold (which is equivalent to the LDF φ). The desired probability can thus be formulated as a multidimensional integral, which can be computed by the efficient algorithm given in [25].

Given a *discrete* finite path ς of length k, an LDF φ, and a time-bound T, we define the set of linear constraints $\mathcal{S}$ generated in Alg. 1. In Alg. 1 line 3 generates the set of constraints from each conjunct in formula φ. In line 5 we add one more constraint to ensure that in the interval of time $[0,T]$ we will reach the last state of ς.

Algorithm 1 Generate a set of linear constraints $\mathcal{S}$ induced by φ, ς and T

Require: LDF φ, a path ς of length k and a time-bound T
Ensure: $\mathcal{S}$ = set of linear constraints
1: $\mathcal{S} = \{\varnothing\}$
2: **for** $j \in J$ **do**
3: $\quad \mathcal{S} = \mathcal{S}\bigcup\left\{\sum_{i\in K_j} c_{ji}\cdot\sum_{\substack{0\leq\ell<k \\ \varsigma[\ell]\models ap_{ji}}} x_\ell \leq M_j\right\}$
4: **end for**
5: $\mathcal{S} = \mathcal{S}\bigcup\left\{\sum_{i=0}^{k-1} x_i \leq T\right\}$
6: $\mathcal{S} = \mathcal{S}\bigcup\{x_i > 0\}$ for all x_i
7: **return** $\mathcal{S}$

Example 4 *Let* $\varphi = \int Idle - \frac{1}{3}\int Busy \leq 0 \wedge \int Idle - \frac{1}{4}\int Sleep \leq 0$ *be an LDF,* $\varsigma = s_0 \to s_1 \to s_2 \to s_1 \to s_3$ *and the time-bound* $t=6$. *The set of linear constraints* $\mathcal{S}$ *induced by* ς, φ *and* t *is:*

$$\mathcal{S} = \begin{cases} -\frac{1}{3}\cdot x_0 + x_1 + 0\cdot x_2 + x_3 \leq 0 \\ 0\cdot x_0 + x_1 - \frac{1}{4}\cdot x_2 + x_3 \leq 0 \\ x_0 + x_1 + x_2 + x_3 < 6 \\ x_0, x_1, x_2, x_3 > 0 \end{cases}$$

Lemma 1 *Let* ς *be a finite path of the* CTMC $\mathcal{C}$, φ *be an LDF and* T *a time-bound. Moreover, let* $\mathcal{S}$ *be the set of linear constraints obtained by Alg. 1. Then*

$$\varsigma[x_0,\ldots,x_{n-1}] \models \varphi \wedge \int 1 \leq T \quad \textit{iff} \quad (x_0,\ldots,x_{n-1}) \in \mathcal{S}.$$

We define $\mathrm{Prob}(\varsigma[\mathcal{S}]) := \Pr^{\mathcal{C}}(\{\rho \in Paths^{\mathcal{C}} \mid \exists\ (x_0,\ldots,x_{n-1})\in\mathcal{S}.\ \rho[0..n]\in\varsigma[x_0,\ldots,x_{n-1}] \wedge \rho[0..n]\models\varphi\})$.

Theorem 4 *Let* ς *be a discrete path of the* CTMC $\mathcal{C}$, $\mathcal{C}[\varphi,G]$ *be the* MRM *induced by* $\mathcal{C}$ *and LDF* φ, *and* $\mathcal{S}$ *the set of linear constraints generated by* ς, φ *and time-bound* t. *Then*

$$\Pr^{\mathcal{C}[\varphi,G]}\{\mathbf{Y}(t)\leq\mathbf{y}\mid\varsigma\} = \mathrm{Prob}(\varsigma[\mathcal{S}]).$$

For future use, declare the function $Volume\_int(\varsigma,\mathcal{S})$ which, given a finite discrete path $\varsigma = s_0 \to \cdots \to s_k$ of length k and a set of linear constraints $\mathcal{S}$ over $x_0,\cdots,x_{k-1}$, returns

$$\prod_{i=0}^{k-1} E(s_i)\cdot P(s_i,s_{i+1})\cdot \underbrace{\int\cdots\int_{\mathcal{S}}}_{k} \prod_{i=0}^{k-1} e^{-E(s_i)\tau_i}dx_i. \tag{5}$$

$\mathrm{Prob}(\varsigma[\mathcal{S}])$ equals $Volume\_int(\varsigma,\mathcal{S})$ when $\mathcal{S}$ is generated from Alg. 1.

Approach based on order statistics.

The problem of computing $\Pr\{Y(t) \leq y \mid \varsigma\}$ is reduced to the computation of the distribution of a linear combination of order statistics uniformly distributed in $[0,1]$ in case $|J| = 1$, i.e., we have a single conjuct in LDF φ. This distribution is calculated through the numerically stable method described in [14]. The state rewards of the CTMC will become the coefficients of the order statistics.

Let $[0,t]$ be an interval of time, and n be the number of transitions in $[0,t]$. Given n transitions, we can divide the interval $[0,t]$ to $n+1$ intervals $I_1, \ldots, I_{n+1}$, and we assign an index i to each interval. Thus, if we stay in state s_1 in the first interval I_1 and state s_1 has reward r_1, we assign index 1 to the first interval. We can divide the CTMC into ℓ distinct reward classes. Without loss of generality, the reward classes are ordered such that $r_1 > \ldots > r_\ell$. We declare a vector $\mathbf{k} = (k_1, \ldots, k_\ell)$, where k_i records the number of times a state with reward r_i has been visited (when index i is not used, $k_i = 0$). Let U_i be the sum of the lengths of intervals of index i defined as follows:

$$\begin{aligned} U_1 &= I_1 + \cdots + I_{k_1}, \\ U_2 &= I_{k_1+1} + \cdots + I_{k_1+k_2}, \\ &\vdots \\ U_\ell &= I_{k_1+\cdots+k_{\ell-1}+1} + \cdots + I_{k_1+\cdots+k_\ell}. \end{aligned}$$

Note that $\sum_{i=1}^{\ell} k_i = n+1$. Then, for n transitions, the total accumulated reward yields: $Y(t) = \sum_{i=1}^{\ell} r_i U_i$.

Now the task is to find the probability $\Pr\{Y(t) \leq y \mid \varsigma\}$. We introduce a renumbering that enables us to disregard all indices that have not been used. Let z_1 be the index of the first nonzero k_i, z_2 be the index of the second nonzero k_i, and so on. Let M be the total number of nonzero k_i's. Then we get $Y(t) = \sum_{i=1}^{M} r_{z_i} U_{z_i}$. Let V_j be the j-th order statistic of a set of n independent and identicaly distributed random variables uniform on $[0,t]$. Note that defining $V_\ell = I_1 + \cdots + I_\ell$ we can re-express each U_i in terms of V_j. More specifically $U_1 = V_{k_1}, U_2 = V_{k_1+k_2} - V_{k_1}, \ldots, U_\ell = t - V_{k_1+\cdots+k_{\ell-1}}$. Rearranging the terms and defining $n_j = \sum_{i=1}^{j} k_i$ for $j = 1, \ldots, \ell-1$, we obtain $Y(t) = \sum_{j=1}^{\ell-1} (r_j - r_{j+1}) V_{n_j} + r_\ell t$. Finally, we get $\Pr\{Y(t) \leq y \mid \varsigma\} = \Pr\left\{\sum_{j=1}^{l-1} (r_j - r_{j+1}) V_{n_j} \leq y - r_\ell t\right\}$.

We can use the algorithm described in [14] to compute the distribution of order statistics uniformly distributed on $[0,1]$, by normalizing with respect to t.

Example 5 *Let $\mathcal{C}$ be the* CTMC *in Fig. 1 with reward structure* $\mathbf{r} = (1, -1, 0, 0)$ *corresponding to the LDP formula $\varphi = \int Busy - \int Idle \leq 0$ and ς be the discrete path $\varsigma = s_0 \to s_1 \to s_0 \to s_1 \to s_3$. In order to calculate* $\Pr\{Y(t) \leq 0 \mid \varsigma\}$ *we define I_i as the time spent in $\varsigma[i]$ for $i \in \{1, \ldots, 5\}$. Let $Y(t)$ be the accumulated reward at time t. The task is to compute* $\Pr\{Y(t) \leq 0 \mid \varsigma\}$. *The accumulated reward is given by $Y(t) = -1 \cdot (I_2 + I_4) + 0 \cdot I_5 + 1 \cdot (I_1 + I_3)$. For every $i \in \{1, \ldots, 5\}$ we introduce a new variable ${I'}_i$ such that $I_1' = I_2$, $I_2' = I_4$, $I_3' = I_5$, $I_4' = I_1$ and $I_5' = I_3$. We obtain a decreasing order for vector* $\mathbf{r}$ *as follows:* $1 > 0 > -1$. *It is clear that we get three reward classes, i.e. $\ell = 3$. We define the vector* $\mathbf{r}' = (-1, 0, 1)$*, which is the vector of the reward classes. Let the vector* $\mathbf{k} = (2, 1, 2)$ *record the number of times a state with reward class r_i' ($i \in \{1, 2, 3\}$) is visited. Let $V_j = \sum_{k=1}^{j} I_k'$ for $1 \leq j \leq 5$. Each V_j is an uniformly distributed variable in $[0,t]$. We can express the accumulated reward in terms of order statistics as follows: $Y(t) = r_3' \cdot V_2 + r_2' \cdot (V_3 - V_2) + r_1' \cdot (V_5 - V_3)$.*

3.1.3 Algorithm

In order to compute $F_s^{s'}(t, \mathbf{y})$ we must pick a finite set $\mathcal{P}$ of paths from $Paths^{\mathcal{D}}$. Following [12], we introduce a threshold $w \in (0,1)$ such that if $\mathrm{Prob}(\varsigma) > w$ then $\varsigma \in \mathcal{P}$. We also fix a maximum length N for the paths in $\mathcal{P}$. Now we define $\mathcal{P}(s, s', w, n) := \{\varsigma \in Paths^{\mathcal{D}} \mid |\varsigma| = n, \varsigma[0] = s, \varsigma[n] = s', \mathrm{Prob}(\varsigma) > w\}$. We can approximate $F_s^{s'}(t, \mathbf{y})$ as

$$\widetilde{F^w_N}_s^{s'}(t, \mathbf{y}) = \sum_{n=0}^{N} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \sum_{\varsigma \in \mathcal{P}(s,s',w,n)} \mathrm{Prob}(\varsigma) \Pr\{\mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\},$$

where w and N must be chosen as stated in Thm 5.

The approximation algorithm to compute $Prob = F_s^{s'}(t, \mathbf{y})$ is given in Alg. 2.

Algorithm 2 Compute $\widetilde{F^w_N}_s^{s'}(t, \mathbf{y})$

$Prob = 0$
$Paths = \{s\}$
while $Paths \neq \varnothing$ **do**
  choose $\varsigma \in Paths$
  $Paths = Paths \setminus \{\varsigma\}$
  if $\mathrm{Prob}(\varsigma) > w$ and $|\varsigma| \leq N$ **then**
    if $\varsigma[|\varsigma|] = s'$ **then**
      $Prob += e^{-\Lambda t} \frac{(\Lambda t)^{|\varsigma|}}{|\varsigma|!} \mathrm{Prob}(\varsigma) \Pr\{\mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\}$
    end if
    for all $s'' \in S$ **do**
      insert $(\varsigma \circ s'')$ into $Paths$
    end for
  end if
end while
return $Prob$

Note that $\circ$ represents the concatenation operator; $\varsigma[|\varsigma|]$ is the last state of ς.

Error Bound.

We give a bound for the truncation of the infinite sum to a finite one, considering only the discrete paths whose probability is greater than w.

Theorem 5 *Given $\varepsilon > 0$, for $N > \Lambda t e^2 + \ln\left(\frac{1}{\varepsilon}\right)$, and $w < \frac{\varepsilon}{\sum_{n=0}^{N} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!}}$, we have* $\left| F_s^{s'}(t, \mathbf{y}) - \widetilde{F^w_N}_s^{s'}(t, \mathbf{y}) \right| \leq 2\varepsilon$.

Complexity.

We analyze the complexity of Alg. 2. Recall that $|S|$ the number of states of $\mathcal{C}$. Alg. 2 is composed of two main steps:

(1) find all paths of length at most N; and (2) for each of those paths ς, compute $\Pr\{\mathbf{Y}(t) \leq \mathbf{y} \mid \varsigma\}$.

Theorem 6 *The complexity of Alg. 2 is $\mathcal{O}(|S|^N \cdot N^{|J|-1})$ using the linear constraint based approach, and $\mathcal{O}(|S|^N \cdot N^2)$ using the order statistics based approach.*

3.2 Unbounded Verification of EDP

In this section we show how to compute $\mathrm{Prob}(\mathcal{C} \models^G \varphi)$. The main idea is that we approximate $\mathrm{Prob}(\mathcal{C} \models^G \varphi)$ by $\mathrm{Prob}(\mathcal{C} \models^G_T \varphi)$ for a sufficiently large $T \in \mathbb{R}_{\geq 0}$. Hence, we reduce the problem to time-bounded verification of EDP, which has been solved in Sect. 3.1. We shall exploit the celebrated Markov inequality. Hence, we first show how to compute the expected time to reach G in $\mathcal{C}$.

Definition 6 *We define a random variable $T_G : Paths^{\mathcal{C}} \to \mathbb{R}_{\geq 0}$ that will denote the first entrance time in a state $s \in G$. More specifically, given a path ρ:*

$$T_G(\rho) = \begin{cases} 0 & \forall j \in \mathbb{N}.\ \rho[j] \notin G \\ \sum\limits_{j=0}^{k-1} \rho\langle j\rangle & o/w, \text{ where } k = \min\{l \mid \rho[l] \in G\}. \end{cases}$$

Lemma 2 *The expected first entrance time $\mathbb{E}_s[T_G]$ from any state $s \in G$ to reach G can be characterized by the following system of linear equations: $\mathbb{E}_s[T_G] = \frac{\mathrm{Prob}(s,\Diamond G)}{E(s)} + \sum\limits_{s' \in S} \mathbf{P}(s,s')\mathbb{E}_{s'}[T_G]$ if $s \notin G$, 0 otherwise, where $\mathrm{Prob}(s,\Diamond G)$ is defined in Def. 2.*

Now we can state the main result of this section.

Theorem 7 $\mathrm{Prob}(\mathcal{C} \models^G \varphi) - \mathrm{Prob}(\mathcal{C} \models^G_T \varphi) \leq \sum\limits_{s\in S} \alpha(s) \frac{\mathbb{E}_s[T_G]}{T}$.

Thanks to this theorem, given an error bound ε and a set of goal states G, we can pick a time bound T such that $T \geq \sum_{s\in S} \alpha(s)\frac{\mathbb{E}_s[T_G]}{\varepsilon}$ and compute $\mathrm{Prob}(\mathcal{C} \models^G_T \varphi)$.

Remark 3 *Here we use Markov inequality. Alternatively one could use the Chebyshev's inequality, which would sharpen Thm. 7 and hence allow a relatively smaller T, at a cost of computing the variance of T_G (instead of the expectation). We choose the current formulation for simplicity.*

4. VERIFICATION OF IDP

In this section, we tackle IDP w.r.t. $\Phi = \int 1 \leq T \to \bigwedge\limits_{j\in J}(\sum\limits_{k\in K_j} c_{jk} \int ap_{jk} \leq M_j)$. As highlighted in Sect. 2, we shall distinguish two cases according to whether T is finite or infinite. First, we give some definitions and algorithms that are common to both cases.

Given an LDF φ, a *discrete* finite path ς of length k and a time-bound T, we define the set of linear constraints $\mathcal{S}$ as in Alg. 3. Note that here $\mathcal{S}$ is different from the one obtained from Alg. 1.

Lemma 3 *Let ς be a finite path of the* CTMC *$\mathcal{C}$, φ be an LDF and t a time-bound. Moreover, let $\mathcal{S}$ be the set of linear constraints obtained by Alg. 3. Then*

$$\varsigma[x_0,\ldots,x_{n-1}] \models^\star \varphi \wedge \int 1 \leq T \quad \textit{iff} \quad (x_0,\ldots,x_{n-1}) \in \mathcal{S}.$$

Algorithm 3 Generate a set of linear constraints $\mathcal{S}$ induced by φ, ς and T

Require: LDF φ, a path ς of length k and a time-bound T
Ensure: $\mathcal{S}$ = set of linear constraints
$\mathcal{S} = \{\varnothing\}$
for $z = 0;\ z < k;\ z{+}{+}$ **do**
  for $j \in J$ **do**
    $\mathcal{S} = \mathcal{S} \bigcup \left\{ \sum\limits_{i\in K_j} c_{ji} \cdot \sum\limits_{\substack{0\leq \ell \leq z \\ \varsigma[\ell] \models ap_{ji}}} x_\ell \leq M_j \right\}$
  end for
end for
$\mathcal{S} = \mathcal{S} \bigcup \left\{ \sum\limits_{i=0}^{k-1} x_i \leq T \right\}$
$\mathcal{S} = \mathcal{S} \bigcup \{x_i > 0\}$ for all x_i
return $\mathcal{S}$

We define $\mathrm{Prob}^\star(\varsigma[\mathcal{S}]) := \Pr^{\mathcal{C}}(\{\rho \in Paths^{\mathcal{C}} \mid \exists\ (x_0,\ldots,x_{n-1}) \in \mathcal{S}.\ \rho[0..n] \in \varsigma[x_0,\ldots,x_{n-1}] \wedge \rho[0..n] \models^\star \varphi\})$, which can be computed by the function $Volume\_int(\varsigma, \mathcal{S})$ (cf. Eq. (5)), where $\mathcal{S}$ is the set of constraints generated from Alg. 3.

Given an infinite timed path ρ, we write $\rho \models^\star_{G,T} \varphi$ if there is some $n \in \mathbb{N}$ such that (1) $\rho[n] \in G$ and $\sum_{i=0}^n \rho\langle i\rangle \leq T$, and (2) for each $0 \leq i \leq n$, $\sum_{j=0}^i \rho\langle j\rangle \leq T$, $\rho[0..i] \models \varphi$. Our task now is to approximate the probability $\boxed{\mathrm{Prob}(\mathcal{C} \models^\star_{G,T} \varphi)}$. For this purpose, we define Alg. 4 that computes an approximation $\widetilde{\mathrm{Prob}}_N(\mathcal{C} \models^\star_{G,T} \varphi)$ of $\mathrm{Prob}(\mathcal{C} \models^\star_{G,T} \varphi)$.

Algorithm 4 Compute $\widetilde{\mathrm{Prob}}_N(\mathcal{C} \models^\star_{G,T} \varphi)$

Require: A CTMC $\mathcal{C}$, an LDF formula φ, set of goal states G, time-bound T, and N
for all $\varsigma \in Paths^{\mathcal{D}}$ s.t. $\exists i.\varsigma[i] \in G$ and $|\varsigma| \leq N$ **do**
  Generate $\mathcal{S}$ from φ, ς, and T, by Alg. 3
  Prob+ $= Volume\_int(\varsigma, \mathcal{S})$
end for
return Prob

4.1 Unbounded Verification of IDP

We are interested in computing $\mathrm{Prob}(\mathcal{C} \models^\star \varphi)$.

Definition 7 *[BSCC] Assume a* CTMC *$\mathcal{C}$. A set of states $B \subseteq S$ is a* strongly connected component *(SCC) of $\mathcal{C}$ if, for any two states $s, s' \in B$, there exists a path $\varsigma = s_0 \to s_1 \to \ldots \to s_n$ such that $s_i \in B$ for $0 \leq i \leq n$, $s_0 = s$ and $s_n = s'$. An SCC B is a* bottom strongly connected component *(BSCC) if no state outside B is reachable from any state in B.*

Definition 8 *Given a BSCC B of the* CTMC *$\mathcal{C}$ and an LDF φ, we say B is* bad *w.r.t. j-th conjunct in φ, φ_j, if $\exists s \in B.\ \exists i \in K_j.\ ap_{ji} \in L(s) \wedge c_{ji} > 0$; otherwise B is* good. *We say B is* good *w.r.t. φ (written $B \models \varphi$) if B is good for each conjunct of φ; otherwise B is* bad *(written $B \not\models \varphi$).*

Lemma 4 *Given a* CTMC *$\mathcal{C} = (S, \mathrm{AP}, L, \alpha, \mathbf{P}, E)$, an LDF φ and a BSCC B we have that, if B is good, then $\Pr^{\mathcal{C}}\{\{\rho \mid \rho \models^\star \varphi\} \mid \Diamond B\} = 1$; and, if B is bad, then $\Pr^{\mathcal{C}}\{\{\rho \mid \rho \models^\star \varphi\} \mid \Diamond B\} = 0$.*

Definition 9 *Given a* CTMC $\mathcal{C} = (S, \mathrm{AP}, L, \alpha, \mathbf{P}, E)$ *and an LDF φ, we define a new* CTMC $\mathcal{C}^a = (S, \mathrm{AP}^a, L^a, \alpha, \mathbf{P}^a, E)$ *as follows:* $\mathrm{AP}^a = \mathrm{AP} \cup \{\perp\}$*, where $\perp$ is fresh; for every good BSCC $B \subseteq S$ and $s \in B$ make s absorbing and let $L^a(s) = L(s) \cup \{\perp\}$; for all other states $s \in S \backslash B$ and $s' \in S$, $\mathbf{P}^a(s, s') = \mathbf{P}(s, s')$, $L^a(s) = L(s)$.*

Example 6 *As an example consider the* CTMC $\mathcal{C}$ *from Fig. 2 (left), in which there are two BSCCs $B_1 = \{s_4, s_5\}$ and $B_2 = \{s_1, s_2, s_3\}$. Moreover, assume that $B_1 \not\models \varphi$ and $B_2 \models \varphi$ for a given LDF φ. After applying Def. 9 to $\mathcal{C}$ we get $\mathcal{C}^a$ shown on the right, where the labels of the states s_1, s_2 and s_3 are augmented with the label $\{\perp\}$ and all the other labels are left unchanged.*

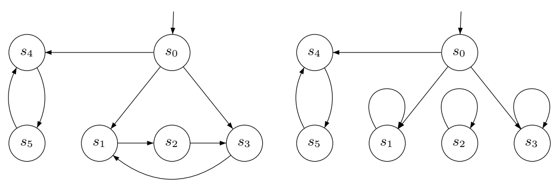


Figure 2: Example BSCC.

We write $\rho \models^\star_G \varphi$ if there exists some $n \in \mathbb{N}$ such that (1) $\rho[n] \in G$, and (2) for each $0 \le i \le n$, $\rho[0..i] \models \varphi$.

Proposition 2 *Given a* CTMC $\mathcal{C} = (S, \mathrm{AP}, L, \alpha, \mathbf{P}, E)$ *and an LDF φ, we have that* $\mathrm{Prob}(\mathcal{C} \models^\star \varphi) = \mathrm{Pr}^{\mathcal{C}^a}(\{\rho \mid \rho \models^\star_G \varphi\})$*, where* $G = \{s \in S \mid \perp \in L(s)\}$.

4.1.1 Algorithm

Algorithm 5 Compute $\widetilde{\mathrm{Prob}}(\mathcal{C} \models^\star \varphi)$

Require: A CTMC $\mathcal{C}$, an LDF formula φ, ε_1 and ε_2
1: Identify all BSCCs B in $\mathcal{C}$
2: $G = \{\varnothing\}$
3: $\mathrm{Prob} = 0$
4: **for each** BSCC B **do**
5:   **if** $B \models \varphi$ **then**
6:     Make every state in B absorbing
7:     $G = G \cup B$
8:   **end if**
9: **end for**
10: Compute $\sum_{s \in S} \alpha(s) \mathbb{E}_s[T_G]$
11: Choose $T > \sum_{s \in S} \alpha(s) \frac{\mathbb{E}_s[T_G]}{\varepsilon_1}$ and $N \ge \Lambda T e^2 + \ln(\frac{1}{\varepsilon_2})$
12: $\mathrm{Prob} = \widetilde{\mathrm{Prob}}_N(\mathcal{C} \models^\star_{G,T} \varphi)$
13: **return** Prob

Alg. 5 computes $\widetilde{\mathrm{Prob}}(\mathcal{C} \models^\star \varphi)$ which is an approximation of $\mathrm{Prob}(\mathcal{C} \models^\star \varphi)$. Lines 4-9 obtain $\mathcal{C}^a$ and the goal states G, according to Def. 9, then the algorithm calls the function $\widetilde{\mathrm{Prob}}_N(\mathcal{C} \models^\star_{G,T} \varphi)$, by choosing T and N, according to the specified error bounds ε_1 and ε_2 respectively.

Error Bound.

Intuitively, there are two factors that contribute to the error introduced by Alg. 5:

- the error introduced by approximating $\mathrm{Pr}^{\mathcal{C}^a}(\{\rho \mid \rho \models^\star_G \varphi\})$ by $\mathrm{Prob}(\mathcal{C}^a \models^\star_{G,T} \varphi)$, which can be obtained in a similar way to Thm. 7. We denote it by ε_1;
- the error introduced through approximating $\mathrm{Prob}(\mathcal{C}^a \models^\star_{G,T} \varphi)$ by $\widetilde{\mathrm{Prob}}_N(\mathcal{C}^a \models^\star_{G,T} \varphi)$. We denote it by ε_2.

Theorem 8 *Given ε_1 and ε_2, we have that*

$$\mathrm{Prob}(\mathcal{C} \models^\star \varphi) - \widetilde{\mathrm{Prob}}(\mathcal{C} \models^\star \varphi) \le \varepsilon_1 + \varepsilon_2.$$

where $\widetilde{\mathrm{Prob}}(\mathcal{C} \models^\star \varphi)$ is given in Alg. 5.

Remark 4 *Given ε* a priori*, one practical way is to let $\varepsilon_1 = \varepsilon_2 = \frac{\varepsilon}{2}$, and hence $T = 2 \sum_{s \in S} \alpha(s) \frac{\mathbb{E}_s[T_G]}{\varepsilon}$ and $N = 2 \sum_{s \in S} \alpha(s) \mathbb{E}_s[T_G] \frac{\Lambda e^2}{\varepsilon} + ln(\frac{4}{\varepsilon})$ suffice.*

4.2 Time-bounded Verification of IDP

In this section we show how to deal with the time-bounded variant of IDP. Given an infinite timed path ρ, we write $\rho \models^{\star,G}_T \varphi$ if $\rho \models^\star \varphi$ and $\rho@T \in G$. The following theorem plays a pivotal role.

Theorem 9 *Given a* CTMC $\mathcal{C}$ *and an LDF* Φ *we have*

$$\mathrm{Prob}(\mathcal{C} \models^\star \Phi) = \sum_{s \in S} \mathrm{Prob}(\mathcal{C} \models^{\star,\{s\}}_T \varphi).$$

The solution boils down to the computation of $\mathrm{Prob}(\mathcal{C} \models^{\star,\{s\}}_T \varphi)$ for each state s. It follows that we compute the approximation $\widetilde{\mathrm{Prob}}(\mathcal{C} \models^\star \Phi)$ by bounding the lenghts of the paths, as shown in Alg. 6. We have the following error bound.

Algorithm 6 Compute $\widetilde{\mathrm{Prob}}(\mathcal{C} \models^\star \Phi)$

Require: A CTMC $\mathcal{C}$, an LDF Φ and ε
$\mathrm{Prob} = 0$
Chose $N \ge \Lambda T e^2 + \ln(\frac{|S|}{\varepsilon})$
for all $s \in S$ **do**
  for all $\varsigma \in \mathit{Paths}^{\mathcal{D}}$ s.t. $\exists n. \varsigma\,[n] = s$ and $|\varsigma| \le N$ **do**
    $\mathcal{S} = \{\varnothing\}$
    for $z = 0$; $z < |\varsigma|$; $z{+}{+}$ **do**
      for $j \in J$ **do**
$$\mathcal{S} = \mathcal{S} \bigcup \left\{ \sum_{i \in K_j} c_{ji} \cdot \sum_{\substack{0 \le \ell \le z \\ \varsigma[\ell] \models ap_{ji}}} x_\ell \le M_j \right\}$$
      end for
    end for
$$\mathcal{S} = \mathcal{S} \bigcup \left\{ \sum_{i=0}^{n} x_i = T \right\}$$
    $\mathcal{S} = \mathcal{S} \bigcup \{x_i > 0\}$ for all x_i
    $\mathrm{Prob}{+}{=} \mathit{Volume\_int}(\varsigma, \mathcal{S})$
  end for
end for
return Prob

Theorem 10 *Given ε and $N \in \mathbb{N}$, it holds that:*

$$\mathrm{Prob}(\mathcal{C} \models^\star \Phi) - \widetilde{\mathrm{Prob}}(\mathcal{C} \models^\star \Phi) < \varepsilon\,.$$

5. CONCLUSION

We have studied the problem of verifying CTMCs against linear durational properties. We focused on two classes of

LDPs, namely, eventuality duration properties and invariance duration properties. The central question we solved is, what is the probability of the set of infinite timed paths of the CTMC which satisfy the given LDP? We presented different algorithms to approximate these probabilities up to a given precision, stating their complexity and error bounds. The implementation of algorithms presented in this paper in PRISM is in progress.

As future work, we plan to study algorithmic verification of more complex duration properties, for instance *response* and *persistence*, as in [6]. It is also interesting to study specifications combining duration properties and temporal properties (in traditional real-time logics, e.g., MTL). The verification of these specifications would be challenging. Extending the current work to CTMDPs is another possible direction.

6. REFERENCES

[1] R. Alur, C. Courcoubetis, and T. A. Henzinger. Computing accumulated delays in real-time systems. In *Formal Methods in System Design*, 11(2):137–155, 1997.

[2] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. On the logical characterisation of performability properties. In *ICALP'00*, LNCS 1853, pp. 780–792. Springer, 2000.

[3] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. In *IEEE Trans. Software Eng.*, 29(6):524–541, 2003.

[4] B. Barbot, T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Efficient CTMC model checking of linear real-time objectives. In *TACAS'11*, LNCS 6605, pp. 128–142. Springer, 2011.

[5] U. Boker, K. Chatterjee, T. A. Henzinger, and O. Kupferman. Temporal specifications with accumulative values. In *LICS'11*, pp. 43–52. IEEE, 2011.

[6] A. Bouajjani, R. Echahed, and J. Sifakis. On model checking for real-time properties with durations. In *LICS'93*, pp. 147–159. IEEE, 1993.

[7] P. Bouyer, U. Fahrenberg, K. G. Larsen, and N. Markey. Timed automata with observers under energy constraints. In *HSCC'10*, pp. 61–70. ACM, 2010.

[8] T. Chen, M. Diciolla, M. Z. Kwiatkowska, and A. Mereacre. Time-bounded verification of CTMCs against real-time specifications. In *FORMATS'11*, LNCS 6919, pp. 26–42. Springer, 2011.

[9] T. Chen, M. Diciolla, M. Kwiatkowska, and A. Mereacre. *Verification of linear duration properties over conitnuous time Markov chains*. Technical report RR-12-02, Department of Computer Science, University of Oxford, 2012.

[10] T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Quantitative model checking of continuous-time Markov chains against timed automata specifications. In *LICS'09*, pp. 309–318. IEEE, 2009.

[11] T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Model checking of continuous-time Markov chains against timed automata specifications. In *Logical Methods in Computer Science*, 7(1–2):1–34, 2011.

[12] L. Cloth. *Model Checking Algorithms for Markov Reward Models*. PhD thesis, University of Twente, The Netherlands, 2006.

[13] M. H. A. Davis. *Markov Models and Optimization*. Chapman and Hall, 1993.

[14] M. C. Diniz, E. de Souza e Silva, and H. R. Gail. Calculating the distribution of a linear combination of uniform order statistics. In *INFORMS Journal on Computing*, 14(2):124–131, 2002.

[15] M. Gribaudo. *Hybrid Formalism for Performance Evaluation: Theory and Applications*. PhD thesis, Università di Torino, 2002.

[16] D. P. Guelev and D. V. Hung. Reasoning about QoS contracts in the probabilistic duration calculus. In *Electr. Notes Theor. Comput. Sci.*, 238(6):41–62, 2010.

[17] B. R. Haverkort, L. Cloth, H. Hermanns, J.-P. Katoen, and C. Baier. Model checking performability properties. In *DSN'02*, pp. 103–112. IEEE, 2002.

[18] G. Horton, V. G. Kulkarni, D. M. Nicol, and K. S. Trivedi. Fluid Stochastic Petri Nets: Theory, Applications, and Solution Techniques. In *European Journal of Operational Research*, 105(1):184–201, 1998.

[19] D. V. Hung and M. Zhang. On verification of probabilistic timed automata against probabilistic duration properties. In *RTCSA'07*, pp. 165–172. IEEE, 2007.

[20] D. V. Hung and C. Zhou. Probabilistic duration calculus for continuous time. In *Formal Asp. Comput.*, 11(1):21–44, 1999.

[21] A. Jensen. Markoff chains as an aid in the study of Markoff processes. In *Skand. Aktuarietidskrift*, 36:87–91, 1953.

[22] Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Decidable integration graphs. In *Inf. Comput.*, 150(2):209–243, 1999.

[23] M. Kwiatkowska, G. Norman, and D. Parker. Stochastic model checking. In *SFM'07*, LNCS 4486, pp. 220–270. Springer, 2007.

[24] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *CAV'11*, LNCS 6806, pp. 585–591. Springer, 2011.

[25] J. B. Lasserre and E. S. Zeron. A Laplace transform algorithm for the volume of a convex polytope. In *J. ACM*, 48(6):1126–1140, 2001.

[26] X. Li, D. V. Hung, and T. Zheng. Checking hybrid automata for linear duration invariants. In *ASIAN'97*, LNCS 1345, pp. 166–180. Springer, 1997.

[27] M. Neuts. *Matrix-Geometric solutions in stochastic models: An algorithimic approach*. John Hopkins University Press, 1981.

[28] Y. Nievergelt. *Foundations of logic and mathematics: applications to computer science and cryptography*. Springer, 2002.

[29] G. Norman, D. Parker, M. Kwiatkowska, S. Shukla, and R. Gupta. Using probabilistic model checking for dynamic power management. In *Formal Aspects of Computing*, 17(2):160–176, 2005.

[30] Q. Qiu, Q. Qu, and M. Pedram. Stochastic modeling of a power-managed system-construction and optimization. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(10):1200 –1217, 2001.

[31] P. H. Thai and D. V. Hung. Verifying linear duration constraints of timed automata. In *ICTAC'04*, LNCS 3407, pp. 295–309. Springer, 2004.

[32] M. Zhang, D. V. Hung, and Z. Liu. Verification of linear duration invariants by model checking CTL properties. In *ICTAC'08*, LNCS 5160, pp. 395–409. Springer, 2008.

[33] C. Zhou, C. A. R. Hoare, and A. P. Ravn. A calculus of durations. In *Inf. Process. Lett.*, 40(5):269–276, 1991.

[34] C. Zhou, J. Zhang, L. Yang, and X. Li. Linear duration invariants. In *FTRTFT'94*, LNCS 863, pp. 86–109. Springer, 1994.