Time-Bounded Verification of CTMCs against Real-Time Specifications [⋆]

Taolue Chen, Marco Diciolla, Marta Kwiatkowska, and Alexandru Mereacre

Department of Computer Science, Oxford University,
Wolfson Building, Parks Road, Oxford, OX1 3QD, United Kingdom

Abstract. In this paper we study time-bounded verification of a finite continuous-time Markov chain (CTMC) $\mathcal{C}$ against a real-time specification, provided either as a metric temporal logic (MTL) property φ, or as a timed automaton (TA) $\mathcal{A}$. The key question is: what is the probability of the set of timed paths of $\mathcal{C}$ that satisfy φ (or are accepted by $\mathcal{A}$) over a time interval of fixed, bounded length? We provide approximation algorithms to solve these problems. We first derive a bound N such that timed paths of $\mathcal{C}$ with at most N discrete jumps are sufficient to approximate the desired probability up to ε. Then, for each discrete (untimed) path σ of length at most N, we generate timed constraints over variables determining the residence time of each state along σ, depending on the real-time specification under consideration. The probability of the set of timed paths, determined by the discrete path and the associated timed constraints, can thus be formulated as a multidimensional integral. Summing up all such probabilities yields the result. For MTL, we consider both the continuous and the pointwise semantics. The approximation algorithms differ mainly in constraints generation for the two types of specifications.

1 Introduction

Verification of *continuous-time Markov chains* (CTMCs) has received much attention in recent years [8]. Thanks to considerable improvements of algorithms, (symbolic) data structures and abstraction techniques, CTMC model checking has emerged as a valuable analysis technique. Aided by powerful software tools, it has been adopted by researchers from, e.g., systems biology, queuing networks and dependability.

The focus of CTMC model checking has primarily been on checking stochastic versions of the *branching-time* temporal logic CTL, such as CSL [7]. The verification of LTL properties reduces to applying well-known algorithms [33,18] to embedded discrete-time Markov chains (DTMCs). Linear-time properties equipped with timing constraints have only recently been considered. In particular, [16,17] treat linear *real-time* specifications that are given as *deterministic timed automata* (DTA). These include properties of the form, "what is the probability to reach a given target state within the deadline, while avoiding unsafe states and not staying too long in any of the dangerous states on the way?". Such properties cannot be expressed in CSL nor in its dialects [6,19]. Model checking DTA properties can be done by a reduction to computing the

[⋆] This work is supported by the ERC Advanced Grant VERIWARE.

reachability probability in a *piecewise deterministic Markov process*, based on the product construction between the CTMC and DTA [17,11]. It remains a challenge to tackle more general real-time specifications like *Metric Temporal Logics* ([4,24], MTL), or *nondeterministic Timed Automata* (TA, [1]). The main difficulty lies in the fact that one cannot easily define a stochastic process out of the CTMC and the MTL formula (or TA), due to the inherent nondeterminism arising from these specifications. The obstacle is somehow fundamental, as it is known that *deterministic* TA are lacking expressiveness compared to their nondeterministic variants or MTL.

Recently, we have seen increasing emphasis on *timed-bounded verification* [27]. Here, "time-bounded" means restricting the modeling and verification efforts to some bounded interval of time, which itself can be taken as a parameter. In verification, queries are phrased over time intervals of fixed, bounded duration. Note that, differently from bounded model checking, which restricts the total number of allowable events (called discrete jumps in this paper), time-bounded verification restricts the total duration under consideration, but *not* the number of events, which can still be unboundedly large owing to the density of time.[1] Instances of time-bounded verification have been considered in the context of stochastic and/or real-time systems [30,9,23,20] and recently studied systematically [27,22]; see [29] for an introduction, where it is argued that the restriction on total duration is very natural for real-time systems.

Inspired by this recent progress, we study the time-bounded verification problem of a CTMC $\mathcal{C}$, against a real-time specification provided as either an MTL formula φ, or as a TA $\mathcal{A}$. The key question is: what is the probability of the set of timed paths of $\mathcal{C}$ that satisfy φ (or are accepted by $\mathcal{A}$) over a fixed time interval $[0, T]$ where $T \in \mathbb{R}_{>0}$? We provide approximation algorithms to solve these problems. Given any $\varepsilon > 0$ a priori, we first derive a bound N such that it is sufficient only to consider timed paths of $\mathcal{C}$ with at most N discrete jumps to approximate the desired probability up to ε. Then, for each *discrete* (untimed) path σ of $\mathcal{C}$ of length at most N, we generate a family of linear constraints, $\mathcal{S}$, over variables determining the residence time of each state in σ. The discrete path σ, together with the associated timing constraints $\mathcal{S}$, determines a set of *timed* paths of $\mathcal{C}$, each of which satisfies φ (or is accepted by $\mathcal{A}$). The probability of this set of timed paths can be formulated as a multidimensional integral, which can be calculated by Laplace transforms, together with an application of the inclusion-exclusion principle. Summing up all such probabilities yields the desired result. Notice that, in the current paper, we consider both the *continuous* and the *pointwise* semantics of MTL (see, e.g. [14]). The approximation algorithms differ mainly in constraints generation for different types of specifications. The family of *linear* constraints are desirable, since we can apply the efficient algorithm for computing the volumes of convex polyhedra [25]. For MTL under the pointwise semantics and TA specifications, constraint generation is relatively easy, while for MTL under the continuous semantics it is more involved. To this end, we first derive constraints in terms of first-order theory of $(\mathbb{R}, +, -, 0, 1, \leq)$, then the Fourier-Motzkin elimination procedure [31, pp.155-156] is

[1] Readers should note that we later bound the number of discrete jumps as an *approximation technique*. This owes to the definition of CTMCs and is irrelevant to the original definition of time-bounded verification.

applied to obtain desired linear constraints. We believe these results are of independent interest, as they have potential usage in domains such as runtime verification.

The approach we take in this paper is quite different from existing results in the literature. Known results can only deal with simpler real-time properties, or are based on *deterministic* property specifications (e.g. DTA). Our technique is based on path exploration of CTMCs, together with a novel analytic methodology to reduce computing the probabilities to a multi-dimensional integral over convex polyhedra. To the best of our knowledge, this is the first work addressing verification of CTMCs against MTL formulas or non-deterministic timed automata.

Related work. Model checking CTMCs against linear real-time specifications has received scant attention so far. To our knowledge, this issue has only been (partially) addressed in [16,6,19]. Baier *et al.* [6] define the logic asCSL where path properties are characterized by (time-bounded) regular expressions over actions and state formulas. The truth value of path formulas depends not only on the available actions in a given time interval, but also on the validity of certain state formulas in intermediate states. asCSL is strictly more expressive than CSL [6]. Model checking asCSL is performed by representing the regular expressions as finite-state automata, followed by computing time-bounded reachability probabilities in the product of CTMC $\mathcal{C}$ and this automaton. In CSL$^{\mathrm{TA}}$ [19], time constraints of until modalities are specified by a single-clock DTA; the resulting logic is at least as expressive as asCSL [19]. The combined behavior of $\mathcal{C}$ and the DTA $\mathcal{A}$ is interpreted as a Markov renewal process, and model checking CSL$^{\mathrm{TA}}$ is reduced to computing the reachability probabilities in a DTMC whose transition probabilities are given by subordinate CTMCs.

Due to space restriction, all the proofs are omitted in the current paper. We refer the readers to [15] for the full proofs, more explanation, and examples.

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 $\mathcal{H}$. Let $Distr(\mathcal{H})$ denote the set of probability measures on this measurable space.

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; $\alpha \in Distr(S)$ is the initial distribution; $\mathbf{P} : S \times S \to [0,1]$ is a stochastic matrix; and $E : S \to \mathbb{R}_{\geq 0}$ is the exit rate function.*

In a CTMC $\mathcal{C}$, state residence times are *exponentially* distributed. More precisely, the residence time X of a state $s \in S$ is a random variable governed by a nonnegative exponential distribution with parameter $E(s)$ (written as $X \sim \mathrm{Exp}(E(s))$). Hence, the probability to exit state s in t time units (t.u. for short) is given by $\int_0^t E(s) \cdot e^{-E(s)\tau} d\tau$. Furthermore, the probability to take the transition from s to s' in t t.u. equals $\mathbf{P}(s,s') \cdot \int_0^t E(s) \cdot e^{-E(s)\tau} d\tau$.

Definition 2. *Given a* CTMC $\mathcal{C} = (S, \mathrm{AP}, L, \alpha, \mathbf{P}, E)$*, we define the following notions.*

- *A (finite)* discrete *path* $\sigma = s_0 \to s_1 \to s_2 \to \ldots$ *is a (finite) sequence of states; we define* σ_i *to be the state* s_i*, and* σ^i *to be the prefix of length* i *of* σ*.*
- *A (finite)* timed *path* $\rho = s_0 \xrightarrow{x_0} s_1 \xrightarrow{x_1} s_2 \xrightarrow{x_2} \ldots$*, where* $x_i \in \mathbb{R}_{>0}$ *for each* $i \geq 0$*, is a sequence starting in state* s_0*; we define* $|\rho|$ *to be the length of a* finite *timed path* ρ*;* $\rho[n] := s_n$ *is the* n*-th state of* ρ *and* $\rho\langle n\rangle := x_n$ *is the time spent in state* s_n*; let* $\rho@t$ *be the state occupied in* ρ *at time* $t \in \mathbb{R}_{\geq 0}$*, i.e.* $\rho@t := \rho[n]$*, where* n *is the smallest index such that* $\sum\limits_{i=0}^{n} \rho\langle i\rangle \geq t$*.*
- *Given a finite discrete path* $\sigma = s_0 \to s_1 \to \cdots \to s_{n-1}$ *of length* n *and* $x_0, \ldots, x_{n-1} \in \mathbb{R}_{>0}$*, define* $\sigma[x_0, \ldots, x_{n-1}]$ *to be the finite timed path* ρ *such that* $\rho[i] := s_i$ *and* $\rho\langle i\rangle := x_i$ *for each* $0 \leq i < n$*.*
- *Let* Γ *be the set of* n*-tuples* $(x_0, \ldots, x_{n-1}) \in \mathbb{R}^n_{>0}$*, then* $\sigma[\Gamma] = \{\sigma[x_0, \ldots, x_{n-1}] \mid (x_0, \ldots, x_{n-1}) \in \Gamma\}$*.*
- *Given a finite (resp. infinite) discrete path* σ *and a finite (resp. infinite) timed path* ρ*, we say* σ *is the* skeleton *of* ρ *if for each* $i \geq 0$*,* $\sigma_i = \rho[i]$*. We write* $\mathbb{S}(\rho)$ *for the skeleton of* ρ*, and for a set of (finite or infinite) timed paths* Ξ*, we write* $\mathbb{S}(\Xi) = \{\mathbb{S}(\rho) \mid \rho \in \Xi\}$*.*
- *Given a* finite *discrete path* σ*, we define* $C_d(\sigma) = \{\sigma\sigma' \mid \sigma' \text{ is an infinite discrete path}\}$ *to be the set of all infinite discrete paths with the same common prefix* σ*.*

Intuitively, a timed path ρ suggests that the CTMC $\mathcal{C}$ starts in state s_0 and stays in this state for x_0 t.u., and then jumps to state s_1, staying there for x_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 $\rho@4 = \rho[1] = s_1$.

Let $\mathit{Paths}^{\mathcal{C}}$ denote the set of infinite timed paths in the CTMC $\mathcal{C}$, and $\mathit{Paths}^{\mathcal{C}}(s)$ the set of infinite timed paths in $\mathcal{C}$ that start in s. Given a time bound $T \in \mathbb{R}_{\geq 0}$ and $N \in \mathbb{N} \cup \{\infty\}$, we define $\mathit{Paths}^{\mathcal{C}}_{T,<N}(s) = \big\{\rho \in \mathit{Paths}^{\mathcal{C}}(s) \mid \exists k.0 \leq k \leq N-1 \text{ and } \sum_{i=0}^{k} \rho\langle i\rangle \geq T\big\}$, to be the set of all timed paths with *at most* $N-1$ discrete jumps in time interval $[0, T]$; and $\mathit{Paths}^{\mathcal{C}}_{T,\geq N}(s) = \big\{\rho \in \mathit{Paths}^{\mathcal{C}}(s) \mid \exists k.0 \leq k \leq N-1, \text{ and } \sum_{i=0}^{k} \rho\langle i\rangle \leq T\big\}$, to be the set of all timed paths with *at least* N jumps in $[0, T]$.

For notational simplicity we will omit the superscript $\mathcal{C}$ when appropriate and also we write $\mathit{Paths}^{\mathcal{C}}_{T}$ instead of $\mathit{Paths}^{\mathcal{C}}_{T,\leq\infty}$ for the set of all timed paths with an arbitrary number of jumps in $[0, T]$. The definition of a Borel space on timed paths through CTMCs follows [7]. A CTMC $\mathcal{C}$ yields a probability measure $\Pr^{\mathcal{C}}$ on $\mathit{Paths}^{\mathcal{C}}$ 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 $\rho \in \mathit{Paths}(s_0)$ such that $\rho[i] = s_i$ $(i \leq k)$, and $\rho\langle i\rangle \in I_i$ $(i < k)$. $\mathcal{F}(\mathit{Paths}(s_0))$ is the smallest σ-algebra on $\mathit{Paths}(s_0)$ 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 $(0 \leq i < k)$ where $I_0, \ldots, I_{k-1}$ range over all sequences of nonempty intervals in $\mathbb{R}_{\geq 0}$. The probability measure $\Pr^{\mathcal{C}}$ on $\mathcal{F}(\mathit{Paths}(s_0))$ is the unique measure defined by induction on k by $\Pr^{\mathcal{C}}(C(s_0)) = \alpha(s_0)$ and for $k > 0$:

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

In general, computing the probability of a cylinder set with k intervals $I_0 \ldots I_{k-1}$ (i.e. k discrete jumps) reduces to calculating k integrals over $I_0 \ldots I_{k-1}$.

2.2 Metric Temporal Logic

Definition 3 (Syntax of MTL). *Let* AP *be an arbitrary nonempty, finite set of atomic propositions. Let $I = [a, b]$ be an interval such that $a, b \in \mathbb{N} \cup \{\infty\}$. The* Metric Temporal Logic *is inductively defined as: $\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \mathcal{U}_I \varphi_2$, where $p \in$* AP *and φ_1, φ_2 are* MTL *formulas.*

We introduce two time-bounded semantics for MTL, as follows.

Definition 4 (Continuous Semantics). *Given an* MTL *formula φ, a time bound T, a timed path ρ and a variable $t \in \mathbb{R}_{\geq 0}$, the satisfaction relation $(\rho, t) \models^c_T \varphi$ is inductively defined as follows:*

$$\begin{array}{ll}(\rho, t) \models^c_T p & \Leftrightarrow p \in L(\rho@t) \wedge t \leq T \\ (\rho, t) \models^c_T \neg\varphi_1 & \Leftrightarrow (\rho, t) \not\models^c_T \varphi_1 \\ (\rho, t) \models^c_T \varphi_1 \wedge \varphi_2 & \Leftrightarrow (\rho, t) \models^c_T \varphi_1 \wedge (\rho, t) \models^c_T \varphi_2 \\ (\rho, t) \models^c_T \varphi_1 \mathcal{U}_I \varphi_2 & \Leftrightarrow \exists t'.\ t \leq t' \leq T \textit{ s.t. } t' - t \in I \wedge (\rho, t') \models^c_T \varphi_2 \wedge \\ & \quad\ \forall t''.\ t \leq t'' < t' \Rightarrow (\rho, t'') \models^c_T \varphi_1\end{array}$$

where $p \in$ AP *and φ_1, φ_2 are* MTL *formulas.*

Definition 5 (Pointwise Semantics). *Given an* MTL *formula φ, a time bound T, a timed path ρ and $i \in \mathbb{N}$, the satisfaction relation $(\rho, i) \models^p_T \varphi$ is inductively defined as follows:*

$$\begin{array}{ll}(\rho, i) \models^p_T p & \Leftrightarrow p \in L(\rho[i]) \wedge \sum_{k=0}^{i} \rho\langle k\rangle \leq T \\ (\rho, i) \models^p_T \neg\varphi_1 & \Leftrightarrow (\rho, i) \not\models^p_T \varphi_1 \\ (\rho, i) \models^p_T \varphi_1 \wedge \varphi_2 & \Leftrightarrow (\rho, i) \models^p_T \varphi_1 \wedge (\rho, i) \models^p_T \varphi_2 \\ (\rho, i) \models^p_T \varphi_1 \mathcal{U}_I \varphi_2 & \Leftrightarrow \exists i'.\ i \leq i' \textit{ s.t. } \sum_{k=i}^{i'} \rho\langle k\rangle \in I \wedge (\rho, i') \models^p_T \varphi_2 \wedge \\ & \quad\ \forall i''.\ i \leq i'' < i' \Rightarrow (\rho, i'') \models^p_T \varphi_1\end{array}$$

where $p \in$ AP*, φ_1, φ_2 are* MTL *formulas and $i', i'' \in \mathbb{N}$.*

2.3 Timed Automata

Let $\mathcal{X} = \{x_1, \ldots, x_p\}$ be a set of nonnegative real-valued variables called *clocks*. An $\mathcal{X}$-valuation is a function $\eta : \mathcal{X} \to \mathbb{R}_{\geq 0}$ assigning to each variable $x \in \mathcal{X}$ a nonnegative real value $\eta(x)$. Let $\mathcal{V}(\mathcal{X})$ denote the set of all valuations over $\mathcal{X}$. A *clock constraint* on $\mathcal{X}$, denoted by g, is a conjunction of expressions of the form $x \bowtie c$ for $x \in \mathcal{X}$,

$\bowtie \in \{<, \leq, >, \geq\}$ and $c \in \mathbb{N}$. Let $\mathcal{B}(\mathcal{X})$ denote the set of clock constraints over $\mathcal{X}$. An $\mathcal{X}$-valuation η *satisfies* constraint $x \bowtie c$, denoted $\eta \models x \bowtie c$, if and only if $\eta(x) \bowtie c$; it satisfies a conjunction of such expressions if and only if η satisfies all of them. Let **0** denote the valuation that assigns 0 to all clocks. For a subset $X \subseteq \mathcal{X}$, the reset of X, denoted $\eta[X := 0]$, is the valuation η' such that $\forall x \in X.\ \eta'(x) := 0$ and $\forall x \notin X.\ \eta'(x) := \eta(x)$. For $\delta \in \mathbb{R}_{\geq 0}$ and $\mathcal{X}$-valuation η, $\eta + \delta$ is the $\mathcal{X}$-valuation η'' such that $\forall x \in \mathcal{X}.\ \eta''(x) := \eta(x) + \delta$, which implies that all clocks proceed at the same speed.

Definition 6 (TA). *A* timed automaton *is a tuple* $\mathcal{A} = (\Sigma, \mathcal{X}, Q, q_0, Q_F, \rightarrow)$ *where* Σ *is a finite* alphabet*;* $\mathcal{X}$ *is a finite set of* clocks*;* Q *is a non empty finite set of* locations *with initial location* $q_0 \in Q$*;* Q_F *is a set of* final *locations; the relation* $\rightarrow \subseteq Q \times \Sigma \times \mathcal{B}(\mathcal{X}) \times \mathbf{2}^{\mathcal{X}} \times Q$ *is an* edge relation.

We refer to $q \xrightarrow{a,g,X} q'$ as an *edge*, where $a \in \Sigma$ is an input symbol, the *guard* g is a clock constraint on the clocks of $\mathcal{A}$, X is the set of clocks that must be reset and q' is the successor location. Intuitively, the edge $q \xrightarrow{a,g,X} q'$ asserts that the TA $\mathcal{A}$ can move from location q to location q' when the input symbol is a and the guard g holds, while the clocks in X should be reset when entering q'. In case no guard is satisfied in a location for a given clock valuation, time can progress. For the sake of simplicity we omit invariants from the definition of TAs. However, the results presented here can be easily extended to TAs enhanced with invariants.

Definition 7. *Given a timed automaton $\mathcal{A}$, we define the following notions.*

- *A* discrete *path of $\mathcal{A}$ is a sequence of states* $w = q_0 \rightarrow q_1 \ldots \rightarrow q_n \cdots$ *where each* $q_i \in Q$*.*
- *A* timed *path of $\mathcal{A}$ is of the form* $\theta = q_0 \xrightarrow{a_0,t_0} q_1 \xrightarrow{a_1,t_1} \ldots q_{n-1} \xrightarrow{a_{n-1},t_{n-1}} q_n \cdots$ *such that* $\eta_0 = \mathbf{0}$*, and for all* $i \geq 0$*,* $a_i \in \Sigma$ *and it holds* $t_i > 0$*,* $\eta_i + t_i \models g_i$ *where* g_i *is the guard on the i-th transition,* $\eta_{i+1} = (\eta_i + t_i)[X_i := 0]$*, where* η_i *is the clock evaluation when entering* q_i*. We say that* θ is accepting *if there exists some* $n \geq 0$ *s.t.* $q_n \in Q_{\mathbf{F}}$.

Definition 8 (Time-bounded Acceptance). *Assume a* CTMC $\mathcal{C} = (S, \mathrm{AP}, L, s_0, \mathbf{P}, E)$ *and a* TA $\mathcal{A} = (2^{AP}, \mathcal{X}, Q, q_0, Q_{\mathbf{F}}, \rightarrow)$*. A* CTMC *timed path* $\rho = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \ldots$*, is accepted by $\mathcal{A}$ if there* exists $n \in \mathbb{N}_{>0}$ *and a corresponding* TA *finite path:* $\theta = q_0 \xrightarrow{L(s_0),t_0} q_1 \xrightarrow{L(s_1),t_1} \ldots q_{n-1} \xrightarrow{L(s_{n-1}),t_{n-1}} q_n$*, such that* $q_n \in Q_F$ *and* $\sum_{i=0}^{n-1} t_i \leq T$*. We write* $\rho \models_T \mathcal{A}$ *to denote that the* CTMC *timed path ρ is accepted by $\mathcal{A}$.*

Remark 1. It is possible that a single CTMC timed path corresponds to multiple TA accepting paths due to the nondeterminism of TA.

3 A Bound on the Number of Discrete Jumps

In this section, we give a bound on discrete jumps of paths of CTMCs such that, when verifying an MTL formula or TA, one only needs to consider those paths whose discrete jumps number at most N. The intuition is that, for a given time interval $[0, T]$, the probability of the set of timed paths which "jump" very frequently is actually very small. Throughout this section we assume a CTMC $\mathcal{C} = (S, \mathrm{AP}, L, \alpha, \mathbf{P}, E)$.

For any $n \in \mathbb{N}$, we define $V^n(s, x) : S \times \mathbb{R}_{\geq 0} \to [0, 1]$ as follows: $V^0(s, x){=}1$ and

$$V^{n+1}(s, x) = \int_0^x E(s)e^{-E(s)\tau} \cdot \sum_{s' \in S} \mathbf{P}(s, s') \cdot V^n(s', x - \tau)d\tau \ \ .$$

Lemma 1. *For all $N \in \mathbb{N}$, $\Pr^{\mathcal{C}}(Paths^{\mathcal{C}}_{T,\geq N}(s)) = V^N(s, T)$.*

We then show how to bound $V^N(s, T)$ analytically. Given a CTMC $\mathcal{C}$, let $\Lambda = \max_{s \in S} E(s)$ and $\epsilon(T, N) = e^{-\Lambda T} \cdot \left(\sum_{i=N}^{\infty} \frac{(\Lambda T)^i}{i!} \right)$.

Lemma 2. $\epsilon(T, N + 1) = \int_0^T \Lambda e^{-\Lambda \tau} \cdot \epsilon(T - \tau, N)d\tau$.

Combining Lem. 1 and Lem. 2, we obtain the following.

Theorem 1. *Given a* CTMC $\mathcal{C}$*, a time bound T and $N \in \mathbb{N}$, $\Pr^{\mathcal{C}}(Paths^{\mathcal{C}}_{T,\geq N}) \leq \epsilon(T, N)$.*

Proposition 1. *Let $\varepsilon \in \mathbb{R}_{>0}$ and $T \in \mathbb{R}_{\geq 0}$. For any $N \geq \Lambda T e^2 + \ln(\frac{1}{\varepsilon})$ we have that $\epsilon(T, N) < \varepsilon$.*

For instance, given a CTMC $\mathcal{C}$ with 10 states, greatest rate $\Lambda = 100$, error bound $\epsilon = 10^{-2}$ and $T = 1000$, we get that $N \geq 738911$. The maximum number of paths to consider would be 10^N.

Remark 2. Readers who are familiar with Poisson distributions will immediately notice that the bound we obtained is actually the probability that there are at least N Poisson arrivals in an interval of time $[0, T]$, with rate Λ. If the CTMC $\mathcal{C}$ is uniform (i.e., each state of $\mathcal{C}$ has the same exit rate), then one could obtain the bound in a straightforward way. However, for the general case, this cannot be achieved directly. Moreover, we point out here that, in order to verify an MTL formula φ or a TA $\mathcal{A}$, one *cannot* apply the unformization technique, which is used only for transient probability computation.

4 MTL Specifications

In this section we study the problem of model checking CTMCs against MTL properties. Let $\Pr^{\mathcal{C}}_T(\varphi) := \Pr^{\mathcal{C}}(\{\rho \in Paths^{\mathcal{C}}_T \mid (\rho, 0) \models^{c}_{T} \varphi\})$ denote the probability that the CTMC $\mathcal{C}$ satisfies the MTL formula φ, for a given time bound T. Notice that, here the definition of $\Pr^{\mathcal{C}}_T(\varphi)$ is for the continuous semantics of MTL. However, we present algorithms to deal with both continuous and pointwise semantics. Instead of computing $\Pr^{\mathcal{C}}_T(\varphi)$, we give a procedure to compute $\Pr^{\mathcal{C}}_{T,<N}(\varphi) := \Pr^{\mathcal{C}}(Paths^{\mathcal{C}}_{T,<N}(\varphi))$ for sufficiently large N which ensures that $\Pr^{\mathcal{C}}_T(\varphi) - \Pr^{\mathcal{C}}_{T,<N}(\varphi) < \varepsilon$ for arbitrarily small $\varepsilon \in \mathbb{R}_{>0}$. This yields an approximation algorithm. The measurability of the set of $Paths^{\mathcal{C}}_{T,<N}(\varphi) := \{\rho \in Paths^{\mathcal{C}}_{T,<N} \mid (\rho, 0) \models^{c}_{T} \varphi\}$ can be shown as in [32]. Below we present an algorithm to compute $\Pr^{\mathcal{C}}_{T,<N}(\varphi)$. We first give a sketch, and provide the crucial sub-procedures in Sec. 4.1 and Sec. 4.2.

Choose N to get the desired error bound ε. The first step of the algorithm is to choose the smallest N from Prop. 1 such that we get the desired error bound ε.

Compute the product $\mathcal{C} \otimes \mathcal{A}_{\widetilde{\varphi}}$. The basic idea of this step is to exclude those CTMC timed paths which definitely fail φ in order to reduce the number of paths to be analyzed. To this end, we define an LTL formula $\widetilde{\varphi}$ such that, if a discrete path of $\mathcal{C}$ fails $\widetilde{\varphi}$, then any timed path with the discrete path as the skeleton (see Def. 2) must fail φ. This is formally stated in Lem. 3. Notice that since we consider the time-bounded semantics of MTL, we need a variant of acceptance for an infinite discrete word and an LTL formula $\widetilde{\varphi}$, which is given in Def. 9. We then construct an NFA out of $\widetilde{\varphi}$ such that only those finite discrete CTMC paths which are accepted by the NFA are the prefixes of the potential skeletons of timed paths satisfying φ. Then we apply the standard product construction, which suffices to identify those CTMC finite discrete paths analyzed in the next step.

Any MTL formula φ can be transformed into a *positive normal form* containing only two temporal operators: $\mathcal{U}_{[a,b]}$ and $\Box_{[a,b]}$, where $(\rho, t) \models^{\mathrm{c}}_{T} \Box_{[a,b]}\varphi$ iff $\forall t' \in [a,b] \Rightarrow (\rho, t+t') \models^{\mathrm{c}}_{T} \varphi$.

Definition 9 (Bounded Semantics of LTL). *Given an* LTL *formula φ, a* finite *discrete path σ and $i \in \mathbb{N}$, the satisfaction relation $(\sigma, i) \models \varphi$ is inductively defined as follows:*

$$\begin{array}{ll}
(\sigma,i) \models p & \Leftrightarrow p \in L(\sigma_i) \textit{ and } i \le |\sigma| \\
(\sigma,i) \models \neg\varphi_1 & \Leftrightarrow (\sigma,i) \not\models \varphi_1 \\
(\sigma,i) \models \varphi_1 \wedge \varphi_2 & \Leftrightarrow (\sigma,i) \models \varphi_1 \wedge (\sigma,i) \models \varphi_2 \\
(\sigma,i) \models \varphi_1 \mathcal{U} \varphi_2 & \Leftrightarrow \exists i'.\ i \le i' \le |\sigma| \textit{ s.t. } (\sigma,i') \models \varphi_2 \wedge \\
 & \quad\ \forall i''.\ i \le i'' < i' \Rightarrow (\sigma,i'') \models \varphi_1
\end{array}$$

where $p \in$ AP, φ_1, φ_2 are LTL *formulas and $i', i'' \in \mathbb{N}$. For an* infinite *discrete path σ, we define $\sigma \models \varphi$ if there exists some $k \ge 0$ such that the finite discrete path $(\sigma^k, 0) \models \varphi$.*

Given any MTL φ *in positive normal form*, we define an (untimed) LTL formula $\widetilde{\varphi}$ as follows:

$$\begin{array}{ll}
\varphi = p & \Rightarrow \widetilde{\varphi} = p \\
\varphi = \neg p & \Rightarrow \widetilde{\varphi} = \neg p \\
\varphi = \varphi_1 \vee \varphi_2 & \Rightarrow \widetilde{\varphi} = \widetilde{\varphi}_1 \vee \widetilde{\varphi}_2 \\
\varphi = \varphi_1 \wedge \varphi_2 & \Rightarrow \widetilde{\varphi} = \widetilde{\varphi}_1 \wedge \widetilde{\varphi}_2 \\
\varphi = \varphi_1 \mathcal{U}_I \varphi_2 & \Rightarrow \widetilde{\varphi} = \widetilde{\varphi}_1 \mathcal{U} \widetilde{\varphi}_2 \\
\varphi = \Box_I \varphi_1 & \Rightarrow \widetilde{\varphi} = \textsc{true}\, \mathcal{U} \widetilde{\varphi}_1
\end{array}$$

where φ_1 and φ_2 are MTL formulas and $\widetilde{\varphi}_1$ and $\widetilde{\varphi}_2$ are LTL formulas.

Remark 3. In the transformation from the MTL formula φ to LTL formula $\widetilde{\varphi}$ we only define the $\neg$ operator for atomic propositions because φ is already in positive normal form. Notice that we transform $\Box_{[a,b]}\varphi$ into $\textsc{true}\, \mathcal{U} \widetilde{\varphi}$ instead of a seemingly more natural $\Box\varphi$, because otherwise in the next step we would not consider timed paths ρ such that $(\rho, 0) \models \varphi$ while $\mathbb{S}(\rho) \not\models \widetilde{\varphi}$. Such paths do exist. For instance, consider the MTL formula $\Box_{[0,2]}p$ and the path $\rho = s_0 \xrightarrow{2.5} s_1 \cdots$ with $L(s_0) = \{p\}$ and $L(s_1) = \{\neg p\}$. Then $(\rho, 0) \models^{c}_{T} \Box_{[0,2]}p$ and $\mathbb{S}(\rho) \not\models \Box p$ (but $\mathbb{S}(\rho) \models \text{TRUE}\,\mathcal{U} p$ as we defined). To conclude, one cannot transform $\Box_{[a,b]}$ by simply removing the time constraints $[a, b]$.

Lemma 3. *Let φ be an* MTL *formula and ρ be a timed path in $\mathcal{C}$. We have that*

$$(\rho, 0) \models^{c}_{T} \varphi \Rightarrow (\mathbb{S}(\rho), 0) \models \widetilde{\varphi}.$$

As the next step, we construct an NFA $\mathcal{A}_{\widetilde{\varphi}}$ which accepts all the prefixes of infinite paths satisfying the formula $\widetilde{\varphi}$ according to Def. 9. The NFA can be obtained by a minor adaptation of the well-known Vardi-Wolper construction [34]. (See [15] for details.) We then build the product of $\mathcal{C}$ and $\mathcal{A}_{\widetilde{\varphi}}$.

Definition 10 (Product $\mathcal{C} \otimes \mathcal{A}_{\widetilde{\varphi}}$). *Given a* CTMC $\mathcal{C} = (S, \text{AP}, L, s_0, \mathbf{P}, E)$ *and an* NFA $\mathcal{A}_{\widetilde{\varphi}} = (Q, 2^{\text{AP}}, \delta, q_0, F)$ *we define the product $\mathcal{C} \otimes \mathcal{A}_{\widetilde{\varphi}}$ to be the tuple $\mathcal{C} \otimes \mathcal{A}_{\widetilde{\varphi}} = (Loc, l_0, Loc_F, \rightsquigarrow)$ where: $Loc = S \times Q$; $l_0 = \langle s_0, q_0 \rangle$; $Loc_F = S \times F$; $\rightsquigarrow \subseteq Loc \times Loc$ such that*

$$\frac{\mathbf{P}(s, s') > 0 \wedge q \xrightarrow{L(s)} q'}{\langle s, q \rangle \rightsquigarrow \langle s', q' \rangle}.$$

The set of accepted timed paths in $\mathcal{C} \otimes \mathcal{A}_{\widetilde{\varphi}}$ is defined by $\Diamond Loc_F$. Notice that we are only interested in the discrete paths of $\mathcal{C} \otimes \mathcal{A}_{\widetilde{\varphi}}$. Therefore, we do not assign probabilities to the transition relation $\rightsquigarrow$ when computing the product. The product is used to check which discrete paths in the CTMC verify the formula $\widetilde{\varphi}$.

Proposition 2. *For any* CTMC $\mathcal{C}$ *and* NFA $\mathcal{A}_{\widetilde{\varphi}}$, $\mathbb{S}(Paths^{\mathcal{C}}_{T}(\varphi)) \subseteq \{C_d(\sigma) \mid \sigma \in \Diamond Loc_F \downharpoonleft_1\}$*, where $Loc_F \downharpoonleft_1$ is the first component of Loc_F.*

Compute all the discrete paths of $\mathcal{C} \otimes \mathcal{A}_{\widetilde{\varphi}}$ of length at most N and calculate the probabilities.

1. Search the graph $\mathcal{C} \otimes \mathcal{A}_{\widetilde{\varphi}}$ to get all the discrete accepting paths σ of $\mathcal{C}$ of length at most N;
2. Run Alg. 1 on each discrete path σ of length $n \leq N$ to obtain the system of linear inequalities $\mathcal{S}$;
3. Compute the probability of $\sigma[\mathcal{S}]$ (cf. Sec. 4.2);
4. Sum up all the probabilities for each discrete path to obtain $\Pr^{\mathcal{C}}_{T,<N}(\varphi)$.

4.1 Constraints Generation

We describe the Alg. 1 that takes as input a discrete path σ of length n and an MTL formula φ. [2] The algorithm returns a family of linear constraints $\mathcal{S} = \bigvee_{i \in I} \bigwedge_{j \in J_i} c_{ij}$ where c_{ij} is a linear inequality over the set of variables $t_0, \ldots, t_{n-1}$. Given a system of linear constraints $\mathcal{S}$ we define the set of feasible solutions to be the tuples $(x_0, \ldots, x_{n-1}) \in \mathbb{R}^n$ such that $(x_0, \ldots, x_{n-1}) \in \mathcal{S}$.

[2] The algorithm Alg. 1 evaluates the formula φ for the continuous semantics.

Algorithm 1 Constraints generation for continuous semantics

```
Require: A finite discrete path $\sigma$ of length $n > 0$, an MTL formula $\varphi$ and a time bound $T$
Ensure: Family of linear inequalities $\mathcal{S}$ over $t_0, \ldots, t_{n-1}$
  $\mathcal{S}'$ :=Constr_Gen($\sigma$,0,$\varphi$)
  $\mathcal{S}$ :=Fourier_Motzkin($\mathcal{S}'$,$t_0$,...,$t_{n-1}$)
  return $\mathcal{S}$

  Function Constr_Gen($\sigma$,$t$,$\varphi$)
  case($\varphi$) :
    $\varphi = p$                          :   return $\big(\bigvee_{k=0}^{n} p \in L(\sigma_k) \wedge \sum_{i=0}^{k} t_i \geq t \wedge \sum_{i=0}^{k-1} t_i < t\big) \wedge t < T$
    $\varphi = \neg\varphi_1$              :   $\mathcal{S}'$ := $\neg$Constr_Gen($\sigma$,$t$,$\varphi_1$)
    $\varphi = \varphi_1 \wedge \varphi_2$ :   $\mathcal{S}'$ :=Constr_Gen($\sigma$,$t$,$\varphi_1$) $\wedge$ Constr_Gen($\sigma$,$t$,$\varphi_2$)
    $\varphi = \varphi_1 \mathcal{U}_{[a,b]} \varphi_2$ :   $\mathcal{S}'$ := $\exists t'.\big(t \leq t' < T \wedge t'-t \geq a \wedge t'-t < b \wedge$ Constr_Gen($\sigma$,$t'$,$\varphi_2$)
                                                    $\wedge\, \forall t''.\, t \leq t'' < t' \Rightarrow$ Constr_Gen($\sigma$,$t''$,$\varphi_1$)$\big)$
  return $\mathcal{S}'$
```

The negation of the family of linear constraints is defined in the standard way. First, the algorithm executes the function Constr_Gen(σ,0,φ). The result is a set of constraints $\mathcal{S}'$ in first-order theory of $(\mathbb{R}, +, -, 0, 1, \leq)$. Second, the algorithm executes the Fourier-Motzkin procedure in order to eliminate all existential and universal quantifiers. This results in a family of linear constraints containing only the variables $t_0, \ldots, t_{n-1}$.

Theorem 2. *Given a discrete path σ of length n, an* MTL *formula φ and a time bound T, we have that $(\sigma[x_0, \ldots, x_{n-1}], 0) \models_T \varphi$ iff $(x_0, \ldots, x_{n-1}) \in \mathcal{S}$, where $\mathcal{S}$ is returned by Alg. 1.*

Example 1. Let $\mathcal{C}$ be a CTMC and let σ be the following finite discrete path on $\mathcal{C}$: $\sigma = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3$. Let $a, b \in$ AP, let $L(s_0) = \{a\}, L(s_1) = \{a\}, L(s_2) = \{a, b\}, L(s_3) = \{\varnothing\}$ and let $\varphi = a\, \mathcal{U}_{[1,2]} b$. The first step of Alg. 1 consists of computing Constr_Gen(σ,0,φ) which returns the following family of linear constraints $\mathcal{S}'$ (the parenthesis "{" denotes the $\wedge$ between the formulas):

$$\exists t'.\, 0 \leq t' < T \wedge t' \geq 1 \wedge t' < 2 \wedge \left\{ \begin{array}{ll} t_0 + t_1 + t_2 & \geq t' \\ t_0 + t_1 & < t' \end{array} \right. \wedge \tag{1}$$

$$\forall t''.\, 0 \leq t'' < t' \Rightarrow \left(t_0 \geq t'' \vee \left\{ \begin{array}{ll} t_0 + t_1 & \geq t'' \\ t_0 & < t'' \end{array} \right. \vee \left\{ \begin{array}{ll} t_0 + t_1 + t_2 & \geq t'' \\ t_0 + t_1 & < t'' \end{array} \right. \right). \tag{2}$$

The constraints in Eq. (2) can always be verified given the constraints in Eq. (1). Moreover, after the Fourier_Motzkin elimination for t', t'' in $\mathcal{S}'$ we obtain the family of constraints $\mathcal{S}$:

$$\mathcal{S} = \left\{ \begin{array}{ll} t_0 + t_1 & < 2 \\ t_0 + t_1 + t_2 & \geq 1 \end{array} \right. .$$

The system $\mathcal{S}$ can be represented using the matrix notation: $\mathcal{S} := \{\mathbf{t} \in \mathbb{R}^n_{>0} \mid \mathbf{A} \cdot \mathbf{t} \trianglelefteq \mathbf{b}\}$, for a given matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, vector $\mathbf{b} \in \mathbb{R}^m$ and $\trianglelefteq \in \{<, \leq\}$. The notation $\mathbb{R}_{>0}$ stands for the semi-closed interval $(0, \infty) \subset \mathbb{R}$. The matrices $\mathbf{A}$, $\mathbf{t}$ and $\mathbf{b}$ in $\mathcal{S}$ are: $\mathbf{A} \in \mathbb{R}^{2 \times 3}$, $\mathbf{t} \in \mathbb{R}^3_{>0}$ and $\mathbf{b} \in \mathbb{R}^2$. More specifically:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ -1 & -1 & -1 \end{bmatrix} ; \mathbf{t} = \begin{bmatrix} t_0 \\ t_1 \\ t_2 \end{bmatrix} ; \mathbf{b} = \begin{bmatrix} 2 \\ -1 \end{bmatrix} .$$

In Alg. 2 we present a procedure which generates a family of linear constraints from a given MTL formula φ under the pointwise semantics. Notice that we do not need to use the `Fourier_Motzkin` elimination procedure, as the family of constraints obtained from `Constr_Gen(`σ`,`0`,`φ`)` contains no quantifiers.

Algorithm 2 Constraints generation for pointwise semantics

```
Require: A finite discrete path $\sigma$ of length $n > 0$, an MTL formula $\varphi$ and a time bound $T$
Ensure: Family of linear inequalities $\mathcal{S}$ over $t_0, \ldots, t_{n-1}$
  return Constr_Gen($\sigma$,$0$,$\varphi$)

  Function Constr_Gen($\sigma$,$i$,$\varphi$)
  case($\varphi$) :
    $\varphi = p$                          : if $p \in L(\sigma_i)$ return $\sum_{k=0}^{i} t_k \leq T$ else return false
    $\varphi = \neg\varphi_1$               : $\mathcal{S}$ := $\neg$Constr_Gen($\sigma$,$i$,$\varphi_1$)
    $\varphi = \varphi_1 \wedge \varphi_2$   : $\mathcal{S}$ :=Constr_Gen($\sigma$,$i$,$\varphi_1$) $\wedge$ Constr_Gen($\sigma$,$i$,$\varphi_2$)
    $\varphi = \varphi_1 \mathcal{U}_{[a,b]} \varphi_2$ : $\mathcal{S}$ := $\big(\bigvee_{i'=i}^{n}$ Constr_Gen($\sigma$,$i'$,$\varphi_2$) $\wedge\ a \leq \sum_{k=i}^{i'} t_k \leq b \ \wedge$
                                                  $(\bigwedge_{i''=i}^{i'-1}$Constr_Gen($\sigma$,$i''$,$\varphi_1$)$)\big)$
  return $\mathcal{S}$
```

Let $\mathcal{S}$ be the family of linear constraints obtained from Alg. 1 and 2. $\mathcal{S}$ is always defined as a *union* of convex polyhedra in $\mathbb{R}^n$, i.e., $\mathcal{S} = \bigvee_{i \in I} \bigwedge_{j \in J_i} c_{ij}$ where, for each $i \in I$, $\bigwedge_{j \in J_i} c_{ij}$ is a convex polyhedron.

4.2 Computing Probabilities

Given a CTMC $\mathcal{C}$, a discrete path σ of length N and the family of linear constraints $\mathcal{S}(t_0, \ldots, t_{N-1})$ obtained from Alg. 1, the main task of this section is to compute the probability of $\sigma[\mathcal{S}]$, i.e., $\Pr^{\mathcal{C}}(\sigma[\mathcal{S}])$. To this end, we first add more constraints to $\mathcal{S}$, namely, for $\mathcal{S} = \bigvee_{i \in I} \bigwedge_{j \in J_i} c_{ij}$ we obtain

$$\overline{\mathcal{S}} = \bigvee_{i \in I} \left(\bigwedge_{j \in J_i} c_{ij} \wedge (t_0 + \ldots + t_{N-1} > T \wedge t_0 + \ldots + t_{N-2} < T) \wedge \bigwedge_{0 \leq k < N} t_k > 0 \right) .$$

These new constraints are used to ensure that there are *exactly* N discrete jumps during the time interval $[0, T]$, and that each residence time is positive.

Now we have N random variables $t_0, \cdots, t_{N-1}$, corresponding to the residence time of each state σ_i for $i \leq N$. The probability $\Pr^{\mathcal{C}}(\sigma[\overline{\mathcal{S}}])$ is thus formulated as the joint probability $\Pr^{\mathcal{C}}(\overline{\mathcal{S}}(t_0, \cdots, t_{N-1}))$, where $t_i \sim \mathrm{Exp}(E(\sigma_i))$ for each $0 \leq i < N$, and $t_0, \cdots, t_{N-1}$ are bounded by the family of linear constraints $\overline{\mathcal{S}}$. The value of the joint probability can be computed through the following multidimensional integration:

$$\Pr^{\mathcal{C}}(\sigma[\overline{\mathcal{S}}]) = \underbrace{\int \cdots \int}_{N}{}_{\overline{\mathcal{S}}(\tau_0,\ldots,\tau_{N-1})} \prod_{i=0}^{N-1} E(s_i) \cdot P(s_i, s_{i+1}) \times e^{-E(s_i)\tau_i} d\tau_i. \quad (3)$$

Proposition 3 **([21]).** *Consider any family of linear inequalities* $\overline{\mathcal{S}} = \bigvee_{i \in I} \bigwedge_{j \in J_i} c_{ij}$*. For each* $i \in I$*, we can write* $\bigwedge_{j \in J_i} c_{ij}$ *in matrix form* $\mathbf{A_i} \cdot \mathbf{t} \trianglelefteq \mathbf{b_i}$ *where* $\trianglelefteq \in \{<, \leq\}$*, and* $\bigwedge_{j \in J_i} c_{ij}$ *is a polyhedron.*

From Prop. 3, we have that $\overline{\mathcal{S}} = \bigvee_{\ell=0}^{k} C_\ell$ where each $C_\ell = \{\mathbf{t} \in \mathbb{R}^n_{>0} | \mathbf{A}_\ell \cdot \mathbf{t} \trianglelefteq \mathbf{b}_\ell\}$ defines a convex set. In case that the union $\bigvee_{\ell=0}^{k} C_\ell$ is not convex, we use the inclusion-exclusion principle to compute $\Pr^{\mathcal{C}}(\sigma[\overline{\mathcal{S}}])$ as follows:

$$\Pr^{\mathcal{C}}(\sigma[\overline{\mathcal{S}}]) = \sum_{\ell=0}^{k} \Pr^{\mathcal{C}}(\sigma[C_\ell]) - \sum_{i,j:0 \leq i < j \leq k} \Pr^{\mathcal{C}}(\sigma[C_i \wedge C_j]) + \\ \sum_{i,j,h:0 \leq i < j < h \leq k} \Pr^{\mathcal{C}}(\sigma[C_i \wedge C_j \wedge C_h]) - \cdots + (-1)^{k-1} \Pr^{\mathcal{C}}(\sigma[C_0 \wedge \cdots \wedge C_k])$$

Remark 4. In our case, the difference between $<$ and $\leq$ in the constraints is marginal, as they would yield the same probability, which can be seen from Eq. (3).

For an index set $L \subseteq \{0, \ldots, k\}$ we write $D = \bigwedge_{\ell \in L} C_\ell$, where C_ℓ defines a polyhedron. By Prop.3, D defines a polyhedron as well. We rewrite $\Pr^{\mathcal{C}}(\sigma[D])$ as:

$$\begin{aligned} \Pr^{\mathcal{C}}(\sigma[D]) &= \prod_{i=0}^{N-1} E(s_i) \cdot P(s_i, s_{i+1}) \cdot \underbrace{\int \cdots \int}_{N}{}_{D} \prod_{i=0}^{N-1} e^{-E(s_i)\tau_i} d\tau_i \\ &= \prod_{i=0}^{N-1} E(s_i) \cdot P(s_i, s_{i+1}) \cdot \underbrace{\int \cdots \int}_{N}{}_{D} e^{-\boldsymbol{E} \cdot \boldsymbol{\tau}} d\boldsymbol{\tau}, \end{aligned}$$

where $\boldsymbol{E} = [E(s_0), \ldots, E(s_{N-1})]$, $\boldsymbol{\tau} = [\tau_0, \ldots, \tau_{N-1}]$ and $\boldsymbol{E} \cdot \boldsymbol{\tau} = \sum_{i=0}^{N-1} E(s_i) \cdot \tau_i$. We use the algorithm of [25] (Sec. 5) to compute efficiently the multidimensional integral $\int \cdots \int_D e^{-\boldsymbol{E} \cdot \boldsymbol{\tau}} d\boldsymbol{\tau}$ based on the Laplace transform. An example of how to compute the integral $\int \cdots \int_D e^{-\boldsymbol{E} \cdot \boldsymbol{\tau}} d\boldsymbol{\tau}$ for a convex set D is given in [15]. The time complexity of solving the multidimensional integral is $\mathcal{O}(n^m)$, where n is the number of constraints and m is the number of variables in D.

Remark 5. Admittedly, it is costly to apply the inclusion-exclusion principle to compute the probabilities. In the worst case, any union of two components is not convex. Notice that efficient algorithms to decide whether the union of two polyhedra is convex there exist; see e.g. [12].

4.3 Main Algorithm and Correctness

We summarize the time-bounded verification algorithm for a CTMC $\mathcal{C}$ against an MTL formula φ in Alg. 3. Recall that Λ is the maximal exit rate appearing in $\mathcal{C}$.

Algorithm 3 Time-bounded verification of a CTMC $\mathcal{C}$ against an MTL formula φ

```
Require: $\mathcal{C}$, $\varphi$, $T$ and $\varepsilon$
Ensure: $\Pr^{\mathcal{C}}_{T,<N}(\varphi)$
  Choose an integer $N \geq \Lambda T e^2 + \ln(\frac{1}{\varepsilon})$
  Transform $\varphi$ into $\widetilde{\varphi}$ and generate NFA $\mathcal{A}_{\widetilde{\varphi}}$ out of $\widetilde{\varphi}$
  Compute the product $\mathcal{C} \otimes \mathcal{A}_{\widetilde{\varphi}}$
  for each discrete path $\sigma$ of $(\mathcal{C} \otimes \mathcal{A}_{\widetilde{\varphi}}) \downarrow_1$ of length $n < N$ do
    Generate the family of linear constraints $\mathcal{S}(t_0, \ldots, t_{n-1})$ using Alg. 1 (or Alg. 2)
    Calculate the probability $p$ of $\sigma[\mathcal{S}]$
    $\Pr^{\mathcal{C}}_{T,<N}(\varphi) := \Pr^{\mathcal{C}}_{T,<N}(\varphi) + p$
  end for
  return $\Pr^{\mathcal{C}}_{T,<N}(\varphi)$
```

For the correctness, we first note that the error is bounded by $\Pr^{\mathcal{C}}_{T,\geq N}(\varphi)$, which is in turn bounded by the probability of the set of timed paths with at least N discrete jumps in $[0, T]$. Then Lem. 4 yields the bound, as follows.

Lemma 4. *Given a* CTMC $\mathcal{C}$*, an* MTL *formula* φ*, a time bound* T *and* $N \in \mathbb{N}$

$$\Pr^{\mathcal{C}}_{T}(\varphi) - \Pr^{\mathcal{C}}_{T,<N}(\varphi) \leq \epsilon(T, N).$$

Theorem 3. *Alg. 3 computes* $\Pr^{\mathcal{C}}_{T,<N}(\varphi)$.

5 TA Specifications

In this section, we show how the procedure outlined in the previous section can be adapted to verify TA specifications on CTMCs. Formally, we intend to compute $\Pr^{\mathcal{C}}_{T}(\mathcal{A})$ $:= \Pr^{\mathcal{C}}(\{\rho \in \mathit{Paths}^{\mathcal{C}}_{T} \mid \rho \models_T \mathcal{A}\})$. As in the case of MTL specifications, we bound $\Pr^{\mathcal{C}}_{T}(\mathcal{A})$ by $\Pr^{\mathcal{C}}_{T,<N}(\mathcal{A}) := \Pr^{\mathcal{C}}(\mathit{Paths}^{\mathcal{C}}_{T,<N}(\mathcal{A}))$, such that $\Pr^{\mathcal{C}}_{T}(\mathcal{A}) - \Pr^{\mathcal{C}}_{T,<N}(\mathcal{A}) < \varepsilon$ for $\varepsilon > 0$. The measurability of the set of paths $\mathit{Paths}^{\mathcal{C}}_{T,<N}(\mathcal{A}) := \{\rho \in \mathit{Paths}^{\mathcal{C}}_{T,<N} \mid \rho \models_T \mathcal{A}\}$ can be shown as in [17].

5.1 Constraints generation

Alur *et. al.* in [5] show how to, given a discrete path π of TA $\mathcal{A}$, construct a graph $\mathcal{G}$ such that $\mathcal{A}$ has a run over π if and only if $\mathcal{G}$ has no negative cost cycle. The graph $\mathcal{G}$ has exactly n nodes and the number of edges of $\mathcal{G}$ depends on the numbers of guards and invariants in $\mathcal{A}$ (see [5] for details). Each edge $e = (i, j)$ (connecting node i to node j) is labeled with a value c such that $c \in \mathcal{H}$ where

$$\mathcal{H} = \{\ldots -2, -1, 0, 1, 2, \ldots\} \cup \{\ldots -2^-, -1^-, 0^-, 1^-, 2^-, \ldots\} \cup \{-\infty, \infty\}$$

The set $\mathcal{H}$ is used to characterize strict and non-strict constraints in $\mathcal{A}$.

For each discrete path σ of the CTMC $\mathcal{C}$ we define $\mathit{\Pi}_\sigma = \{\pi \mid \pi_i \xrightarrow{L(\sigma_i)} \pi_{i+1}$ for all $0 \leq i \leq n-1\}$.

Theorem 4. *Given a discrete path σ of length n, a* TA *$\mathcal{A}$ and a time bound T, we have that $\sigma[t_0, \ldots, t_{n-1}]$ is accepted by $\mathcal{A}$ iff $(t_0, \ldots, t_{n-1}) \in \mathcal{S}$, where $\mathcal{S}$ is returned by Alg. 4.*

Algorithm 4 Constraints generation for a TA

```
Require: A finite discrete path $\sigma$ of length $n > 0$ and a TA $\mathcal{A}$
Ensure: Family of linear constraints $\mathcal{S}$
For the discrete path $\sigma$ compute the set $\mathit{\Pi}_\sigma$
for each $\pi \in \mathit{\Pi}_\sigma$ do
    Generate the graph $\mathcal{G}$
    $\mathcal{S}_\pi := \varnothing$
    for each edge $e(i,j) \in \mathcal{G}$ labeled with $c$ do
        $\mathcal{S}_\pi := \mathcal{S}_\pi \wedge t_i - t_j < c$
    end for
    $\mathcal{S} := \mathcal{S} \vee \Big(\mathcal{S}_\pi \wedge (t_0 + \ldots + t_{n-1} > T \wedge t_0 + \ldots + t_{n-2} < T) \wedge \bigwedge_{0 \leq k < n} t_k > 0\Big)$
end for
return $\mathcal{S}$
```

5.2 Algorithm for TA

Given a timed automaton $\mathcal{A}$ we write $\bar{\mathcal{A}}$ to denote the NFA obtained by removing all the guards, clocks and invariants from $\mathcal{A}$. The product $\mathcal{C} \otimes \bar{\mathcal{A}}$ follows Def. 10. Similarly to Prop. 2, we have that

Proposition 4. *For any* CTMC *$\mathcal{C}$ and NFA $\bar{\mathcal{A}}$, $\mathbb{S}(\mathit{Paths}^{\mathcal{C}}_{T}(\mathcal{A})) \subseteq \{C_d(\sigma) \mid \sigma \in \Diamond \mathit{Loc}_F \downharpoonleft_1\}$, where Loc_F is the set of final locations in $\mathcal{C} \otimes \bar{\mathcal{A}}$.*

The approximation algorithm for time-bounded verification of a TA specification $\mathcal{A}$ is given in Alg. 5.

Lemma 5. *Given a* CTMC *$\mathcal{C}$, a* TA *specification $\mathcal{A}$, a time bound T and $N \in \mathbb{N}$*

$$\mathrm{Pr}^{\mathcal{C}}_{T}(\mathcal{A}) - \mathrm{Pr}^{\mathcal{C}}_{T,<N}(\mathcal{A}) \leq \epsilon(T, N).$$

Theorem 5. *Alg. 5 computes* $\mathrm{Pr}^{\mathcal{C}}_{T,<N}(\mathcal{A})$.

6 Conclusion

In this paper we have studied time-bounded verification of CTMCs against real-time specifications. In particular, we presented effective procedures to approximate the probability of the set of timed paths of CTMCs that satisfy real-time specifications over a

Algorithm 5 Time-bounded verification of a TA specification $\mathcal{A}$ against a CTMC $\mathcal{C}$

```
Require: $\mathcal{C}$, $\mathcal{A}$, $T$ and $\varepsilon$
Ensure: $\Pr^{\mathcal{C}}_{T,<N}(\mathcal{A})$
Choose an integer $N \geq \Lambda T e^2 + \ln(\frac{1}{\varepsilon})$
for each discrete path $\sigma$ of $(\mathcal{C} \otimes \bar{\mathcal{A}}) \downharpoonright_1$ of length $n < N$ do
    Calculate the family of linear constraints $\mathcal{S}(t_0, \ldots, t_{n-1})$ with Algorithm 4
    Calculate the probability $p$ of $\sigma[\mathcal{S}]$
    $\Pr^{\mathcal{C}}_{T,<N}(\mathcal{A}) := \Pr^{\mathcal{C}}_{T,<N}(\mathcal{A}) + p$
end for
return $\Pr^{\mathcal{C}}_{T,<N}(\mathcal{A})$
```

time interval of fixed bounded length, arbitrarily closely. For the real-time specifications, we focused on MTL under both the continuous and pointwise semantics, and general timed-automata.

The aim of the current paper is to provide effective approximation algorithms. We leave the precise complexity as future work. Notice that, for MTL, the satisfiability problem over CTMCs is undecidable for continuous semantics [2] while it has non-primitive recursive complexity for pointwise semantics [28]. These results do not carry over directly to CTMCs, as they do not involve nondeterminism. Moreover, we mention that since our algorithms involve computation over reals, it might make more sense to consider different computation models (e.g. the BSS model [13]) and the complexity theory therein, rather than the standard Turing model. Notice that one could also apply discretization to solve the problem. However, it is not clear how the probabilities are preserved in the discretized model.

Recently [26] showed that, under the bounded-variability assumption (BVA), an MTL formula can be transformed into a deterministic timed automaton. Roughly, a timed path satisfies the BVA if there exist Δ and k such that, for *every* interval of the form $[t, t + \Delta]$, the number of discrete jumps is at most k. Clearly, this is related to the bound on discrete jumps in $[0, T]$. However, the BVA is a "global" assumption over $[0, \infty)$, so it does not apply to time-bounded verification. Also, it is not clear for us how to bound the error under this assumption. It would be interesting to investigate whether one could obtain a DTA out of MTL under our assumption of finitely many jumps over $[0, T]$, which could yield an alternative way to solve the problem, based on previous work of two authors [17]. A natural question is how to tackle the traditional (time-unbounded) verification. The scheme introduced in this paper still works. However, one cannot guarantee an approximation to stay within the given error bound ε, which means that the resulting procedure is *not* an approximation algorithm any more. It is also interesting to tackle real-time specifications given as *alternating timed automata* [22] or as TPTL formulas [3,10], as they subsume MTL. We claim that the scheme can be applied in a straightforward way. However, one needs new constraints generation procedures. We leave them as future work.

Acknowledgement. We are grateful to Klaus Dräger, Joost-Pieter Katoen, and anonymous referees for fruitful discussions and constructive comments.

References

1. R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.
2. R. Alur, T. Feder, and T. A. Henzinger. The benefits of relaxing punctuality. *J. ACM*, 43(1):116–146, 1996.
3. R. Alur and T. A. Henzinger. A Really Temporal Logic. *J. ACM*, 41(1):181–204, 1994.
4. R. Alur and T. A. Henzinger. Real-time logics: Complexity and expressiveness. In *LICS*, pages 390–401, 1990.
5. R. Alur, R. P. Kurshan, and M. Viswanathan. Membership questions for timed and hybrid automata. In *IEEE Real-Time Systems Symposium*, pages 254–263, 1998.
6. C. Baier, L. Cloth, B. R. Haverkort, M. Kuntz, and M. Siegle. Model checking Markov chains with actions and state labels. *IEEE Trans. Software Eng.*, 33(4):209–224, 2007.
7. C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. Software Eng.*, 29(6):524–541, 2003.
8. C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Performance evaluation and model checking join forces. *Commun. ACM*, 53(9):76–85, 2010.
9. C. Baier, H. Hermanns, J.-P. Katoen, and B. R. Haverkort. Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. *Theor. Comput. Sci.*, 345(1):2–26, 2005.
10. P. Bouyer, F. Chevalier and N. Markey. On the expressiveness of TPTL and MTL. *Inf. Comput.*,208(2):97–116, 2010.
11. B. Barbot, T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Efficient CTMC model checking of linear real-time objectives. In P. A. Abdulla and K. R. M. Leino, editors, *TACAS*, volume 6605 of *Lecture Notes in Computer Science*, pages 128–142. Springer, 2011.
12. A. Bemporad, K. Fukuda, and F. D. Torrisi. Convexity recognition of the union of polyhedra. *Comput. Geom.*, 18(3):141–154, 2001.
13. L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and real computation*. Springer-Verlag, 1998.
14. P. Bouyer. *From Qualitative to Quantitative Analysis of Timed Systems*. Mémoire d'habilitation, Université Paris 7, Paris, France, Jan. 2009.
15. T. Chen, M. Diciolla, M. Kwiatkowska, and A. Mereacre. Time-bounded verification of CTMCs against real-time specifications. Tech. Rep. RR-11-06, Department of Computer Science, University of Oxford, 2011.
16. T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Quantitative model checking of continuous-time Markov chains against timed automata specifications. In *LICS*, pages 309–318, 2009.
17. T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Model checking of continuous-time Markov chains against timed automata specifications. *Logical Methods in Computer Science*, 7(1–2):1–34, 2011.
18. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *J. ACM)*, 42(4):857–907, 1995.
19. S. Donatelli, S. Haddad, and J. Sproston. Model checking timed and stochastic properties with CSL$^{\mathrm{TA}}$. *IEEE Trans. Software Eng.*, 35(2):224–240, 2009.
20. E. M. Hahn, H. Hermanns, B. Wachter, and L. Zhang. Time-bounded model checking of infinite-state continuous-time Markov chains. *Fundam. Inform.*, 95(1):129–155, 2009.
21. J. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I.: Fundamentals*. Springer-Verlag, 1994.
22. M. Jenkins, J. Ouaknine, A. Rabinovich, and J. Worrell. Alternating timed automata over bounded time. In *LICS*, pages 60–69. IEEE Computer Society, 2010.

23. J.-P. Katoen and I. S. Zapreev. Safe on-the-fly steady-state detection for time-bounded reachability. In *QEST*, pages 301–310, 2006.
24. R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.
25. J. B. Lasserre and E. S. Zeron. A Laplace transform algorithm for the volume of a convex polytope. *J. ACM*, 48(6):1126–1140, 2001.
26. D. Nickovic and N. Piterman. From MTL to deterministic timed automata. In *FORMATS*, pages 152–167, 2010.
27. J. Ouaknine, A. Rabinovich, and J. Worrell. Time-bounded verification. In M. Bravetti and G. Zavattaro, editors, *CONCUR*, volume 5710 of *Lecture Notes in Computer Science*, pages 496–510. Springer, 2009.
28. J. Ouaknine and J. Worrell. On the decidability and complexity of metric temporal logic over finite words. *Logical Methods in Computer Science*, 3(1), 2007.
29. J. Ouaknine and J. Worrell. Towards a theory of time-bounded verification. In S. Abramsky, C. Gavoille, C. Kirchner, F. M. auf der Heide, and P. G. Spirakis, editors, *ICALP (2)*, volume 6199 of *Lecture Notes in Computer Science*, pages 22–37. Springer, 2010.
30. O. Roux and V. Rusu. Verifying time-bounded properties for ELECTRE reactive programs with stopwatch automata. *Hybrid Systems II*, pages 405–416, 1995.
31. A. Schrijver. *Theory of linear and integer programming*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1999.
32. A. Sharma and J.-P. Katoen. Weighted lumpability on Markov chains. In *In 8th Ershov Informatics Conference*. LNCS, 2011.
33. M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *FOCS*, pages 327–338, 1985.
34. M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification (preliminary report). In *LICS*, pages 332–344, 1986.