Safe On-The-Fly Steady-State Detection for Time-Bounded Reachability

Joost-Pieter Katoen[a,b] and Ivan S. Zapreev[a,b]

[a]RWTH Aachen, Software Modeling and Verification Group, D-52056 Aachen, Germany

[b]University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Abstract

The time-bounded reachability problem for continuous-time Markov chains (CTMCs) amounts to determine the probability to reach a (set of) goal state(s) within a given time span, such that prior to reaching the goal certain states are avoided. Efficient algorithms for time-bounded reachability are at the heart of probabilistic model checkers such as PRISM and ETMCC. For large time spans, on-the-fly steady-state detection is commonly applied. To obtain correct results (up to a given accuracy), it is essential to avoid detecting premature stationarity. This paper gives a detailed account of criteria for steady-state detection in the setting of time-bounded reachability. This is done for forward and backward reachability algorithms. As a spin-off of this study, new results for on-the-fly steady-state detection during CTMC transient analysis are reported. Based on these results, a precise procedure for steady-state detection for time-bounded reachability is obtained. Experiments show the impact of these results in probabilistic model checking.

1. Introduction

When performing transient analysis for a continuous-time Markov chain (CTMC), it is common practice—in particular in case of large time spans—to use a built-in steady-state detection technique [16, 24]. The underlying idea is to be able to detect whether the CTMC has reached an equilibrium before the finish of the (large) time bound. Whenever such equilibrium is detected, the transient computation can be stopped thus saving expensive computational steps. The criteria for detecting such equilibria when guaranteeing a given overall inaccuracy are, however, not always clear and may lead to the detection of premature equilibria. This may happen, for instance, when the probability mass in the CTMC under consideration only changes slightly in a series of computational steps due to a "slow" movement.

Why is on-the-fly steady-state detection of importance for probabilistic model checking [13]? One of the key issues in the model checking of continuous-time probabilistic models such as CTMCs is the *time-bounded reachability problem.* This entails to determine the probability to reach a (set of) goal state(s) within a given time span, such that prior to reaching the goal certain states are avoided. This corresponds to a probabilistic variant of time-bounded until formulae as they are used in e.g., the verification of timed automata. For CTMCs, the calculation of such probabilities can be reduced to a transient analysis on a modified CTMC [2]. In case of local model checking, a transient analysis needs to be carried out for a single state only. This can be efficiently done in a forward manner, i.e., starting from the state of interest. For global model checking, the validity of a logical property needs to be checked in every state and thus this probability must be computed for all states. Doing so in a backward fashion yields an improvement of $\mathcal{O}(N)$ over-performing the forward algorithm N times,where N is the size of the state space [11].

As checking time-bounded reachability properties reduces to transient analysis, on-the-fly steady-state detection can be exploited in probabilistic model checking. Probabilistic model checkers such as PRISM [15], ETMCC [9] and its variants for stochastic Petri nets (such as GreatSPN [5] and the APNN Toolbox [3]) have adopted this technique for model checking CSL (Continuous Stochastic Logic [1, 2]), a variant of CTL. These model checkers have basically adopted steady-state detection *as it is*, without tailoring it to the specific nature of time-bounded reachability. Other tools that support model checking CSL, such as VESTA [19] and Ymer [25], use a statistical testing approach and do not support steady-state detection.

In this paper, we present a detailed analysis of the use of on-the-fly steady-state detection in this setting. We start by revisiting and slightly sharpening a (well-known) result by Fox-Glynn [6] that is used in computing Poisson probabilities, an essential ingredient in CTMC transient analysis. Based on this result, we prove criteria to safely decide whether an equilibrium has been reached for both the backward and forward reachability algorithm. These criteria sharpen known results for on-the-fly steady-state detection for CTMC transient analysis [16] and for CSL model checking [23, 24]. Based on these theoretical results, a simple procedure is proposed to safely detect equilibria. This is done by exploiting the structure of the CTMC that is obtained when reducing time-bounded reachability to transient analysis. Experimental results complete this paper and show the impact of our theoretical achievements. By means of an artificial, though extremely simple CTMC, we show that various existing probabilistic model checkers detect a premature equilibrium resulting in incorrect verification results. We report similar observations for the workstation cluster [7, 4, 23, 14, 18], an example that has established itself as a benchmark problem for probabilistic model checking. (These results confirm a similar phenomenon reported in a recent analysis of the IEEE 802.11 group communication protocol [17].) Based on these observations, we firmly believe that the results in this paper improve current probabilistic model checking technology.

Organization of the paper. Section 2 introduces relevant concepts of CTMCs and their discrete variant, DTMCs. Section 3 presents the slight refinement of the Fox-Glynn error-bound. Section 4 introduces the time-bounded reachability problem, its forward and backward algorithm, and the use of on-the-fly steady-state detection. Sections 5 and 6 contain the main contribution of this paper; these sections present new criteria for detecting equilibria during time-bounded reachability, and the algorithm to safely detect steady state. Section 7 reports on the conducted experiments, and Section 8 concludes.

2. Markov chain preliminaries

This section recalls stationary probabilities for DTMCs, transient probabilities of CTMCs, and some related concepts. For more information we refer to standard textbooks on Markov chains [22, 8]. Let $S = \{1, \ldots, N\}$ be a finite set of state indexes with $|S| = N$.

2.1. Discrete-time Markov chains

Definition 1 *A Discrete-Time Markov Chain (*DTMC*) is a tuple* $(S,\, \mathcal{P})$ *with finite set* S *of states and state-transition probability matrix* $\mathcal{P} : S \times S \to [0,\, 1]$*, where* $\mathcal{P} = (p_{i,j})$ *and* $\forall i \in S : \sum_{j \in S} p_{i,j} = 1$.

The matrix entry $p_{i,j}$ denotes the probability to move from state i to state j in one step. Let $\overrightarrow{p\,(0)}$ denote the initial probability distribution of $(S,\, \mathcal{P})$, i.e., $p\,(0)_i$ denotes the probability to be initially in state i.

Definition 2 *The* limiting *state-probability of DTMC* $(S,\, \mathcal{P})$ *is a vector* $\overrightarrow{p^*(0)}$ *such that:*

$$\overrightarrow{p^*(0)} = lim_{n \to \infty}\, \overrightarrow{p\,(0)} \cdot \mathcal{P}^n \tag{1}$$

Whenever this limit exists, it can be obtained by solving the system of linear equations:

$$\overrightarrow{p} = \overrightarrow{p} \cdot \mathcal{P},\; \sum_{i \in S} p_i = 1 \tag{2}$$

In case the limit (1) does not exist, the equation (2) may still have a solution. The solution of equation (2) is also known as *stationary* or *steady-state* distribution. In general, the limiting state-probability $p^*(0)_i$ can be interpreted either as the proportion of time the DTMC is in state i on the long run or, alternatively, as the probability that the DTMC is in state i when taking a snapshot after a long time. The vector $\overrightarrow{p}$ in equation (2) is the left eigenvector of $\mathcal{P}$ that corresponds to the unit eigenvalue. As $\mathcal{P}$ is a stochastic matrix, $\mathcal{P}$ always has a unit eigenvalue, and no other eigenvalue exceeds it in modulus.

Theorem 1 *[8] An irreducible and aperiodic finite DTMC has a unique limiting distribution (1). This distribution coincides with the steady-state distribution and does not depend on the initial distribution* $\overrightarrow{p\,(0)}$.

Informally, *irreducible* means that every state is reachable from every other state, (i.e. the underlying graph is strongly connected). A DTMC is periodic if one of its states is periodic. State i is periodic if for some $d > 1$ it holds that the probability to return to state i in n steps is 0, for all n such that $n \;mod\; d \neq 0$. A sufficient condition for an irreducible DTMC to be *aperiodic* is that there exists at least one state with a self loop. A possible way to determine the steady-state probabilities for a DTMC is to compute the dominant (eigenvalue, eigenvector) pair of $\mathcal{P}$. This is described in the following.

The Power method. The Power method [21] is a well-known numerical technique for computing the dominant eigenvalue and its eigenvector. In case of a stochastic matrix $\mathcal{P}$, it amounts to:

$$\overrightarrow{x\,(m)} = \overrightarrow{x\,(m{-}1)}{\cdot}\mathcal{P}$$

For aperiodic $\mathcal{P}$, convergence is guaranteed. If, in addition, $\mathcal{P}$ is irreducible, then the result does not depend on the initial vector $\overrightarrow{x\,(0)}$. According to [21], the number K of iterations required to satisfy a tolerance criterion $\varepsilon > 0$ may be approximately obtained as:

$$K = \frac{\log_2 \varepsilon}{\log_2 |\lambda_2|}$$

where λ_2 is a subdominant eigenvalue of $\mathcal{P}$.

In practice however, λ_2 is difficult to compute, and other convergence tests are used [21], such as for $M > 0$:

1. An absolute convergence test: $\|\overrightarrow{x\,(i)} - \overrightarrow{x\,(i+M)}\|_v < \varepsilon$
2. A relative convergence test: $\max_{j \in 1,..,N} \left(\frac{|x(i+M)_j - x(i)_j|}{|x(i+M)_j|} \right) < \varepsilon$

In general, M is a function of the convergence rate and the iteration index i, but may be constant. Unfortunately, none of these tests gives a precise estimate. Stewart [21] therefore suggests to envisage a *battery* of convergence tests, all of which must be satisfied before the approximation is accepted as being sufficiently accurate.

2.2. Transient and stationary probabilities

Definition 3 *A Continuous-Time Markov Chain (*CTMC*) is a tuple $(S,\ \mathcal{Q})$ with finite set S of states and generator matrix $\mathcal{Q} : S \times S \to \mathcal{R}$, where $\mathcal{Q} = (q_{i,j})$, $\forall i, j \in S : i \neq j : q_{i,j} \geq 0$, and $\forall i \in S : q_{i,i} = -\sum_{j \in S,\, i \neq j} q_{i,j}$.*

Let us explain the intuitive meaning of $q_{i,j}$. In a CTMC, state residence times are exponentially distributed. More precisely, the time spent in state i is governed by a negative exponential distribution with rate $|q_{i,i}|$. The rate $|q_{i,i}|$ thus specifies the total rate to leave state i. On leaving state i, a discrete probabilistic choice takes place among all possible successors, i.e. all states j for which $q_{i,j} > 0$. The probability to move to state j is defined as $q_{i,j}/|q_{i,i}|$. The transient probabilities of CTMC $(S,\ \mathcal{Q})$, with time $t \in \mathcal{R}$, are defined by the following differential equation:

$$\frac{d\overrightarrow{\pi^*\,(0,t)}}{dt} = \overrightarrow{\pi^*\,(0,t)} \cdot \mathcal{Q}$$

The solution of this differential equation system is given by:

$$\overrightarrow{\pi^*\,(0,t)} = \overrightarrow{p\,(0)} \cdot e^{\mathcal{Q} \cdot t} \tag{3}$$

Here, $\overrightarrow{\pi^*\,(0,t)}$ denotes the state-probability after a delay of t time-units given that $\overrightarrow{p\,(0)}$ is the initial distribution. $\pi^*\,(0,t)_i$ thus is the probability to be in state i after t time-units given $\overrightarrow{p\,(0)}$.

The stationary (steady-state) probabilities for a CTMC are a solution of the following system of linear equations:

$$\overrightarrow{p} \cdot \mathcal{Q} = \overrightarrow{0}, \quad \text{where} \quad \sum_{i \in S} p_i = 1 \tag{4}$$

A solution of equation (4) may be found by alternatively presenting it as the following unit eigenvalue problem [20]:

$$\overrightarrow{p} \cdot \mathcal{P} = \overrightarrow{p}$$

where $\mathcal{P} = \frac{\mathcal{Q}}{q} + \mathcal{I}$ and $q \geq \max_{i \in S} |q_{i,i}|$. It is well known [20] that if

$$q > \max_{i \in S} |q_{i,i}|$$

then all eigenvalues, except the unit eigenvalue, are strictly less than unity in modulus, which makes DTMC $\mathcal{P}$ aperiodic.

Jensen's method Jensen's method, also known as uniformization, replaces $\mathcal{Q}$ by $\mathcal{P}$ in equation (3). Expanding the matrix exponent according to Taylor-McLaurin yields:

$$\overrightarrow{\pi^*\,(0,t)} = \sum_{i=0}^{\infty} \gamma_i(t) \cdot \overrightarrow{p(0,i)} \tag{5}$$

where $\gamma_i(t) = e^{-qt} \frac{(qt)^i}{i!}$ is the Poisson density function, $\overrightarrow{p(0,i)} = \overrightarrow{p\,(0)} \cdot \mathcal{P}^i$, and q is the *uniformization rate*.

2.3. On-the-fly steady-state detection

Equation (5) contains $\overrightarrow{p\,(0)} \cdot \mathcal{P}^i$ which is the power iteration for the DTMC $\mathcal{P}$, and that is where the steady-state detection comes into play. Malhotra *et al.* [16] present a numerical method, which takes into account steady-state detection, for computing CTMC transient probabilities (see equation (5)) with an overall error bound ε. For the sake of this paper, we state their result in the following form:

Theorem 2 *[16] Let $(S,\ \mathcal{P})$ be an aperiodic DTMC with initial distribution $\overrightarrow{p\,(0)}$ and steady-state distribution $\overrightarrow{p^*\,(0)}$. If for some K and $\delta > 0$ it holds that $\forall i \geq K : \|\overrightarrow{p^*\,(0)} - \overrightarrow{p(0,i)}\|_v \leq \delta$, where $\|.\|_v$ is an arbitrary vector norm* [1]*, then for*

$$\overrightarrow{\pi^*\,(0,t)} = \sum_{i=0}^{\infty} \gamma_i(t) \overrightarrow{p(0,i)}$$

and for inaccuracy $\varepsilon > 0$:

$$\overrightarrow{\pi\,(0,t)} = \begin{cases} \overrightarrow{p(0,K)} & \textit{, if } K < \mathcal{L}_\epsilon \\ \sum_{i=\mathcal{L}_\epsilon}^{K} \gamma_i(t) \overrightarrow{p(0,i)} + & \\ \overrightarrow{p(0,K)} \left(1 - \sum_{i=0}^{K} \gamma_i(t)\right) & \textit{, if } \mathcal{L}_\epsilon \leq K \leq \mathcal{R}_\epsilon \\ \sum_{i=\mathcal{L}_\epsilon}^{\mathcal{R}_\epsilon} \gamma_i(t) \overrightarrow{p(0,i)} & \textit{, if } K > \mathcal{R}_\epsilon \end{cases} \tag{6}$$

the following inequality holds:

$$\boxed{\|\overrightarrow{\pi^*\,(0,t)} - \overrightarrow{\pi\,(0,t)}\|_v \ \leq \ 2\delta + \frac{\varepsilon}{2}}$$

Here, $\mathcal{L}_\epsilon$ and $\mathcal{R}_\epsilon$ are computed using the Fox-Glynn algorithm (see below), such that $\sum_{i=0}^{\mathcal{L}_\epsilon - 1} \gamma_i(t) \leq \frac{\varepsilon}{2}$, and $\sum_{i=\mathcal{R}_\epsilon + 1}^{\infty} \gamma_i(t) \leq \frac{\varepsilon}{2}$.

[1] It should be noted that Theorem 2 does not hold for an arbitrary norm $\|.\|_v$. In fact, an additional condition is required, that is $\|\overrightarrow{p}\|_v \leq 1$ for any distribution vector $\overrightarrow{p}$.

Theorem 2 can now be used to obtain a criterion for guaranteeing an overall inaccuracy of $\varepsilon > 0$ for transient analysis with on-the-fly steady-state detection.

Corollary 3 *Under the same conditions as Theorem 2:*

$$\boxed{\begin{aligned} &\|\overrightarrow{p^*(0)} - \overrightarrow{p(0,K)}\|_v \leq \frac{\varepsilon}{4} \textit{ implies} \\ &\qquad \|\overrightarrow{\pi^*(0,t)} - \overrightarrow{\pi(0,t)}\|_v \leq \varepsilon \qquad (7) \end{aligned}}$$

As $\|\overrightarrow{p^*(0)} - \overrightarrow{p(0,K)}\|_v$ is not known during computations (since $\overrightarrow{p^*(0)}$ is unknown, and typically not computed a priori as this is computationally too expensive), [16] suggests to use the absolute convergence test. This amounts to replace the premise in equation (7) by:

$$\|\overrightarrow{p(0,i)} - \overrightarrow{p(0,i+M)}\|_v \leq \frac{\varepsilon}{4} \quad \text{for } M > 0$$

Accordingly, $\overrightarrow{p(0,K)}$ with $K = i+M$ is used as an approximation of the real steady-state distribution. This approach thus boils down to comparing probability vectors that are M iterations apart. Once these probability vectors are close enough, it is assumed that the CTMC has reached an equilibrium. This approach, of course, has the drawback that due to the use of an approximation of the stationary probability, an equilibrium may be detected prematurely. A detailed analysis revealed that in deriving the above result in [16], an important ingredient of the Fox-Glynn algorithm is not considered, viz. the so-called weights. (Weights will be discussed in detail in the next section). It will be shown in the remainder of this paper that weights play an important role to obtain *safe* criteria for detecting equilibria.

3. Fox-Glynn error bound revisited

Recall that $\gamma_i(t) = e^{-qt}\frac{(qt)^i}{i!}$ is a Poisson density function, thus $\sum_{i=0}^{\infty}\gamma_i(t) = 1$. $\gamma_i(t)$ denotes the probability that i events occur in a period of t time units, given that the average rate of events is q. The particular shape of the Poisson density function allows for ignoring the "tails" of the density function. For a given error bound $\varepsilon > 0$, these so-called left and right *truncation points* are given by $\mathcal{L}_\epsilon$ and $\mathcal{R}_\epsilon$ such that:

$$\sum_{i=0}^{\mathcal{L}_\epsilon - 1}\gamma_i(t) \leq \frac{\varepsilon}{4}, \text{ and } \sum_{i=\mathcal{R}_\epsilon+1}^{\infty}\gamma_i(t) \leq \frac{\varepsilon}{4} \quad .$$

For real-valued function $f : \mathbb{N} \to \mathbb{R}$, the Fox-Glynn algorithm [6] allows to compute

$$\sum_{i=0}^{\infty}\gamma_i(t)f(i) \approx \frac{1}{W}\sum_{i=\mathcal{L}_\epsilon}^{\mathcal{R}_\epsilon} w_i(t)f(i)$$

where $w_i(t) = \alpha\gamma_i(t)$ with $\mathcal{L}_\epsilon \leq i \leq \mathcal{R}_\epsilon$, for some constant $\alpha \neq 0$, are *weights*, and $W = \sum_{i=\mathcal{L}_\epsilon}^{\mathcal{R}_\epsilon} w_i(t)$ is a normalization weight. Note the resemblance of the left-hand side of the last equation and equation (5). The weights $w_i(t)$ and the normalization weight W are used to prevent underflows and are computed; α is an unknown constant.

Proposition 4 *[6] For real-valued function f with $\|f\| = \sup_{i\in\mathbb{N}}|f(i)|$ and $\sum_{i=\mathcal{L}_\epsilon}^{\mathcal{R}_\epsilon}\gamma_i(t) \geq 1 - \frac{\varepsilon}{2}$ it holds:*

$$\left|\sum_{i=0}^{\infty}\gamma_i(t)f(i) - \frac{1}{W}\sum_{i=\mathcal{L}_\epsilon}^{\mathcal{R}_\epsilon} w_i(t)f(i)\right| \leq \varepsilon \cdot \|f\| \quad .$$

The following refinement can be made for the case when f does not change sign, i.e., $f(i) \leq 0$ or $f(i) \geq 0$, for all i.

Proposition 5 *For real-valued function f that does not change sign with $\|f\| = \sup_{i\in\mathbb{N}}|f(i)|$ and $\sum_{i=\mathcal{L}_\epsilon}^{\mathcal{R}_\epsilon}\gamma_i(t) \geq 1 - \frac{\varepsilon}{2}$ it holds:*

$$\left|\sum_{i=0}^{\infty}\gamma_i(t)f(i) - \frac{1}{W}\sum_{i=\mathcal{L}_\epsilon}^{\mathcal{R}_\epsilon} w_i(t)f(i)\right| \leq \frac{\varepsilon}{2} \cdot \|f\| \quad .$$

Proof For proof details see [12]. □

4. Time-bounded reachability

A recent popular approach to analyze properties of Markov chains is probabilistic model checking; for an overview see [13]. Time-bounded reachability is at the heart of model-checking algorithms for CTMCs. Let us explain this problem by means of an example. Consider a CTMC with many states among which some illegal (or: forbidden) states and some goal states. Suppose we are interested in determining the states from which a goal state may be reached with a high probability, say at least 0.92, within time interval $[0, 14.5]$, while never visiting an illegal state before reaching its goal. We thus consider scenarios in which the system starts in some state $s \in S$, visits any number of states which are not illegal, while finally ending up in some goal state. In the logic CSL [1, 2], a continuous-time probabilistic extension of CTL, this requirement is formulated by:

$$\mathrm{P}_{\geq 0.92}(\mathcal{A}\,\mathrm{U}^{[0,14.5]}\,\mathcal{G})$$

where $\mathcal{A}$ denotes the set of legal (allowed) states and $\mathcal{G}$ denotes the set of goal states.[2]

The part between parentheses characterizes a set of paths, where a single path is an alternating sequence $s_0t_0s_1t_1s_2t_2\ldots$ where $t_j > 0$ denotes the residence time in state s_j, and $s_0 = s$. Such path satisfies $\mathcal{A}\,\mathrm{U}^{[0,14.5]}\,\mathcal{G}$

[2] In logical formulas, we identify set $\mathcal{A}$ with its characteristic function.

if there exists an index $j \geq 0$ such that $s_j \in \mathcal{G}$, $s_i \in \mathcal{A}$ for all $i < j$, and s_j is reached within 14.5 time units, i.e., $\sum_{i=0}^{j-1} t_i \leq 14.5$. For the sake of brevity, we do not present the detailed semantics of this operator; these details can be found in [1, 2].

Time-bounded reachability thus amounts to compute *Prob*$(s,\ \mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G})$, the probability for state s satisfying formula $\mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G}$. This problem can be reduced to the computation of transient probabilities in a modified CTMC [2]. This goes as follows. In the original CTMC $(S,\ \mathcal{Q})$, all states in $\mathcal{G}$ and all states that are neither in $\mathcal{A}$ nor in $\mathcal{G}$ are made absorbing, i.e., their outgoing transitions are removed. This operation is formally defined by:

Definition 4 *For CTMC* $(S,\ \mathcal{Q})$ *and* $S' \subseteq S$ *let CTMC* $(S, \mathcal{Q}')$ *be obtained by making all states in* S' *absorbing, i.e.,* $\mathcal{Q}' = \mathcal{Q}[S']$ *where* $q'_{i,j} = q_{i,j}$ *if* $i \notin S'$ *and 0 otherwise.*

Note, that in order to make a state s in a DTMC absorbing, all outgoing transitions of s are removed and s is equipped with a self-loop (with probability 1). In a CTMC, it suffices to remove the outgoing transitions. It now follows that:

$$\textit{Prob}(s, \mathcal{A}\mathrm{U}^{[0,t]}\mathcal{G}) \text{ in } (S,\ \mathcal{Q}) \ = \ \textit{Prob}(s, S\mathrm{U}^{[t,t]}\mathcal{G}) \text{ in } (S, \mathcal{Q}\left[\mathcal{I} \cup \mathcal{G}\right])$$

where $\mathcal{I} = S \setminus (\mathcal{A} \cup \mathcal{G})$.

For any state $s \in S$, the probability *Prob*$(s,\ \mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G})$ can be computed using Algorithm 1, where the matrix exponent can be computed numerically using uniformization and $\overrightarrow{1_s}$ is the initial distribution for the case when starting in state s. [11] suggests an improvement of this method (cf. Algorithm 2). Like before, the matrix exponent is computed using uniformization. Let $\overrightarrow{i_\mathcal{G}}$ be the characteristic vector of the set $\mathcal{G}$.

Algorithm 1 Computing *Prob*$(s,\ \mathcal{A}\,\mathrm{U}^{[0,t]}\,\mathcal{G})$ in a "forward" manner

```
Determine $\mathcal{Q}\left[\mathcal{I} \cup \mathcal{G}\right]$
Compute $\overrightarrow{\pi^*\left(s,t\right)} = \overrightarrow{1_s} \cdot e^{\mathcal{Q}[\mathcal{I}\cup\mathcal{G}]t}$
Return $Prob(s,\ \mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G}) = \sum_{j \in \mathcal{G}} \pi^*\left(s,t\right)_j$
```

Algorithm 2 Computing *Prob*$(s,\ \mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G})$ in a "backward" manner

```
Determine $\mathcal{Q}\left[\mathcal{I} \cup \mathcal{G}\right]$
Compute $\overrightarrow{\pi^*\left(t\right)} = e^{\mathcal{Q}[\mathcal{I}\cup\mathcal{G}]t} \cdot \overrightarrow{i_\mathcal{G}}$
Return $\forall s \in 1,..,N : Prob(s,\ \mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G}) = \pi^*\left(t\right)_s$
```

4.1. On-the-fly steady state detection

The steady-state detection for transient analysis of CTMCs discussed in Section 2.3, is applicable to the forward computations in a straightforward way. Steady-state detection for backward computations has been recently discussed in [24]. The approach by Younes *et al.* is based on the following result.

Theorem 6 *Let* $(S,\ \mathcal{P})$ *be an aperiodic DTMC with* $Ind \subseteq S$ *such that* $\forall j \in Ind : P(j,j) = 1$, $\overrightarrow{p(i)} = \mathcal{P}^i \cdot \overrightarrow{i_{Ind}}$ *and steady-state vector* $\overrightarrow{p^*}$. *If for some* K *and* $\delta > 0$ *it holds that* $\forall i \geq K : \|\overrightarrow{p^*} - \overrightarrow{p(i)}\|_v \leq \delta$, *then for*

$$\overrightarrow{\pi^*\left(t\right)} = \sum_{i=0}^{\infty} \gamma_i(t)\overrightarrow{p(i)}$$

and for inaccuracy $\varepsilon > 0$*:*

$$\overrightarrow{\pi\left(t\right)} = \begin{cases} \overrightarrow{p(K)} & \textit{, if } K < \mathcal{L}_\epsilon \\ \sum_{i=\mathcal{L}_\epsilon}^{K} \gamma_i(t)\overrightarrow{p(i)} + \\ \overrightarrow{p(K)}\left(1 - \sum_{i=\mathcal{L}_\epsilon}^{K} \gamma_i(t)\right) & \textit{, if } \mathcal{L}_\epsilon \leq K \leq \mathcal{R}_\epsilon \\ \sum_{i=\mathcal{L}_\epsilon}^{\mathcal{R}_\epsilon} \gamma_i(t)\overrightarrow{p(i)} & \textit{, if } K > \mathcal{R}_\epsilon \end{cases} \tag{8}$$

the following inequality holds:

$$\boxed{\|\overrightarrow{\pi^*\left(t\right)} - \overrightarrow{\pi\left(t\right)}\|_v \leq 2\delta + \frac{\varepsilon}{2}}$$

Here $\mathcal{L}_\epsilon$*, and* $\mathcal{R}_\epsilon$ *are computed using the Fox-Glynn algorithm, such that* $\sum_{i=0}^{\mathcal{L}_\epsilon - 1} \gamma_i(t) \leq \frac{\varepsilon}{2}$ *and* $\sum_{i=\mathcal{R}_\epsilon+1}^{\infty} \gamma_i(t) \leq \frac{\varepsilon}{2}$.

In [24], this result has led to the following practical check for steady-state:

$$\boxed{\begin{array}{c} \|\overrightarrow{p^*} - \overrightarrow{p(K)}\|_v \leq \dfrac{\varepsilon}{8} \text{ implies} \\ \forall j \in S : -\dfrac{\varepsilon}{4} \leq \pi^*\left(t\right)_j - \pi\left(t\right)_j \leq \dfrac{3}{4}\varepsilon \qquad (9) \end{array}}$$

As before, since $\overrightarrow{p^*}$ is not known during computations, the absolute convergence test is used instead. That is, the premise is replaced by $\|\overrightarrow{p(i)} - \overrightarrow{p(i+M)}\|_v \leq \frac{\varepsilon}{8}$. The vector $\overrightarrow{p(K)}$ with $K = i{+}M$ is thus used as an approximation of the steady-state vector. Whereas for the forward analysis case, the convergence test bound equals $\frac{\varepsilon}{4}$ (cf. equation (7)), for the backward analysis this is $\frac{\varepsilon}{8}$ (cf. equation (9)). One may question how safe (and tight) this criterion for equilibrium detection is. As these results are based on [16], the drawbacks of this method are inherited. A detailed look at the equations (6) and (8) for the case $\mathcal{L}_\epsilon \leq K \leq \mathcal{R}_\epsilon$ reveals that the second summation for the backward case starts at $i = \mathcal{L}_\epsilon$ rather than $i = 0$. The justification for this change is unclear, but has a non-negligible impact on the bound. To be more precise, this change of the summation index implicitly increases the error bound and this is not taken care of. More importantly, though, the analysis resulting in Theorem 6 is based on the assumption that the steady-state detection error is two-sided, whereas—due to the backward nature of the algorithm— it is in fact *one-sided*.

5. Criteria for steady-state detection

In this section, we provide new error bounds for on-the-fly steady-state detection during time-bounded reachability. These results apply to the forward algorithm, i.e., standard transient analysis, as well as to the backward algorithm.

Remark. The error estimate in [16] is norm based and relies on the *geometrical convergence* of power iterations for an aperiodic DTMC. The geometrical convergence is usually proved, based on the *total variation norm* which, in an N-dimensional space, is the l^{∞}-norm defined as $\|v\|_v^{\infty} = \max_{i \in 1,..,N} |v_i|$. As all norms in a finite-dimensional space are equivalent, the convergence result holds for any chosen norm. The error analysis, however, is vulnerable to the kind of norm used. For example, in the backward case, the $\overrightarrow{i_{\mathcal{G}}}$ vector is not a distribution and $\forall j \in 1,..,N : 0 \leq p(i)_j \leq 1$, where $\overrightarrow{p(i)} = \mathcal{P}^i \cdot \overrightarrow{i_{\mathcal{G}}}$. Thus, for instance, if we have N states and take the Euclidean norm $\|.\|_v^2$, we obtain $\|\sum_{i=0}^{\mathcal{L}_\epsilon - 1} \gamma_i(t)\overrightarrow{p(i)}\|_v^2 \not\leq \frac{\varepsilon}{4}$, but $\|\sum_{i=0}^{\mathcal{L}_\epsilon - 1} \gamma_i(t)\overrightarrow{p(i)}\|_v^2 \leq \frac{\sqrt{N}}{4}\varepsilon$. The error analysis below is done for vector elements and uses the $\|.\|_v^{\infty}$ norm.

5.1. Transient analysis

Let $p^*(0)_j$ be the j'th component of the precise steady-state solution $\overrightarrow{p^*(0)}$, considering forward computations, for the initial distribution $\overrightarrow{p\,(0)}$. Let $\pi^*\,(0,t)_j$ be the j'th component of $\overrightarrow{\pi^*\,(0,t)}$, see equation (5). For the case $\mathcal{L}_\epsilon \leq K \leq \mathcal{R}_\epsilon$ we consider:

$$\overrightarrow{\pi\,(0,t)} = \frac{1}{W}\sum_{i=\mathcal{L}_\epsilon}^{K} w_i(t)\overrightarrow{p(0,i)} + \overrightarrow{p(0,K)}\left(1 - \frac{1}{W}\sum_{i=\mathcal{L}_\epsilon}^{K} w_i(t)\right)$$

This equation is obtained from (6) by replacing the lower bound of the index of the second summation by $i = \mathcal{L}_\epsilon$, as it was done in (8), and assuming the Fox-Glynn algorithm is used for computations. This is where $w_i(t)$ and W play a role.

Theorem 7 *Let $(S, \mathcal{P})$ be an aperiodic DTMC with initial distribution $\overrightarrow{p\,(0)}$, steady-state distribution $\overrightarrow{p^*(0)}$ and $Ind \subseteq S$. If for some K and $\delta > 0$ it holds that $\forall i \geq K : \|\overrightarrow{p^*(0)} - \overrightarrow{p(0,i)}\|_v^{\infty} \leq \delta$ then for*

$$\overrightarrow{\pi^*\,(0,t)} = \sum_{i=0}^{\infty} \gamma_i(t)\overrightarrow{p(0,i)}$$

and for inaccuracy $\varepsilon > 0$:

$$\overrightarrow{\pi\,(0,t)} = \begin{cases} \overrightarrow{p(0,K)} & \text{, if } K < \mathcal{L}_\epsilon \\ \frac{1}{W}\sum_{i=\mathcal{L}_\epsilon}^{K} w_i(t)\overrightarrow{p(0,i)} + \\ \overrightarrow{p(0,K)}\left(1 - \frac{1}{W}\sum_{i=\mathcal{L}_\epsilon}^{K} w_i(t)\right) & \text{, if } \mathcal{L}_\epsilon \leq K \leq \mathcal{R}_\epsilon \\ \frac{1}{W}\sum_{i=\mathcal{L}_\epsilon}^{\mathcal{R}_\epsilon} w_i(t)\overrightarrow{p(0,i)} & \text{, if } K > \mathcal{R}_\epsilon \end{cases}$$

the following inequality holds:

$$\boxed{\left|\sum_{j \in Ind}\left(\pi^*\,(0,t)_j - \pi\,(0,t)_j\right)\right| \leq 2\delta|Ind| + \frac{3}{4}\varepsilon}$$

Here W, $w_i(t)$, $\mathcal{L}_\epsilon$, and $\mathcal{R}_\epsilon$ are computed using the Fox-Glynn algorithm, such that $\sum_{i=0}^{\mathcal{L}_\epsilon - 1}\gamma_i(t) \leq \frac{\varepsilon}{4}$, and $\sum_{i=\mathcal{R}_\epsilon+1}^{\infty}\gamma_i(t) \leq \frac{\varepsilon}{4}$, and $|Ind|$ is the cardinality of Ind.

Proof For proof details see [12]. □

Corollary 8 *Under the same conditions as Theorem 7:*

$$\boxed{\begin{array}{l} \|\overrightarrow{p^*(0)} - \overrightarrow{p(0,K)}\|_v^{\infty} \leq \dfrac{\varepsilon}{8|Ind|} \text{ implies} \\ \qquad \left|\sum_{j \in Ind}\left(\pi^*\,(0,t)_j - \pi\,(0,t)_j\right)\right| \leq \varepsilon \qquad (10) \end{array}}$$

Let us now return to the calculation of time-bounded reachability probabilities. For computing the probability *Prob*$(s,\ \mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G})$ in state s, we have $Ind = \mathcal{G}$ and $\overrightarrow{p\,(0)} = \overrightarrow{1_s}$. According to the above results, the safe stopping criterion to obtain an overall inaccuracy of ε equals $\|\overrightarrow{p^*(s)} - \overrightarrow{p(s,K)}\|_v^{\infty} \leq \frac{\varepsilon}{8|\mathcal{G}|}$. We point out the main differences between our result and the results referred to in Section 2.3. First of all, Theorem 7 takes into account the weights $w_i(t)$ (and the normalization factor W) for determining $\overrightarrow{\pi\,(0,t)}$. Hence, different summation bounds for the case $\mathcal{L}_\epsilon \leq K \leq \mathcal{R}_\epsilon$ (as $w_i(t) = 0$ for all $i < \mathcal{L}_\epsilon$) occur in the definition of $\overrightarrow{\pi\,(0,t)}$. Secondly, due to the refined bound for Fox-Glynn (cf. Proposition 5), the bounds on the left and right truncation errors on which Theorem 7 is based are two times tighter than ($\frac{1}{4}$ instead of $\frac{1}{2}$) the corresponding truncations errors that form the basis for Theorem 2. Theorem 7 refers to the l^{∞}-norm, whereas the norm in Theorem 2 is left implicit.

The resulting steady-state detection criterion

$$\|\overrightarrow{p^*(0)} - \overrightarrow{p(0,K)}\|_v^{\infty} \leq \frac{\varepsilon}{8}$$

for the case $|Ind| = 1$ (the error for a single vector element $\pi\,(0,t)_j$), is tighter than the bound provided in [16] (see equation (7)):

$$\|\overrightarrow{p^*(0)} - \overrightarrow{p(0,K)}\|_v \leq \frac{\varepsilon}{4}$$

The fact that the resulting bound is similar to that in Section 4.1 for backward computations is due to the fact that the weights introduce an additional error. In the next section, it will be shown that for backward computations—even taking into account the error introduced by weights—the steady-state detection criterion is weaker than in [24] .

5.2. Backward computations

Let $\pi^*(t)_j$ be the j'th component of $\overrightarrow{\pi^*(t)}$, and p_j^* be the j'th component of $\overrightarrow{p^*}$. Notice, that, unlike the case for forward computations, $\forall j \in 1,..,N : p_j^* - p(i)_j \geq 0$ for all $i \geq K$, because $\forall j \in 1,..,N, \forall i \geq 0 : p(i)_j \leq p(i+1)_j \leq \overrightarrow{p^*}$.

Theorem 9 *Let $(S,\ \mathcal{P})$ be an aperiodic DTMC with $Ind \subseteq S$ such that $\forall j \in Ind : P(j,j) = 1$, $\overrightarrow{p(i)} = \mathcal{P}^i \cdot \overrightarrow{i_{Ind}}$ and steady-state vector $\overrightarrow{p^*}$. If for some K and $\delta > 0$ it holds that $\forall i \geq K : \forall j \in 1,..,N : 0 \leq p_j^* - p(i)_j \leq \delta$, then for*

$$\overrightarrow{\pi^*(t)} = \sum_{i=0}^{\infty} \gamma_i(t)\overrightarrow{p(i)}$$

and for inaccuracy $\varepsilon > 0$:

$$\overrightarrow{\pi(t)} = \begin{cases} \overrightarrow{p(K)} & \text{, if } K < \mathcal{L}_\epsilon \\ \frac{1}{W}\sum_{i=\mathcal{L}_\epsilon}^{K} w_i(t)\overrightarrow{p(i)} + \\ \overrightarrow{p(K)}\left(1 - \frac{1}{W}\sum_{i=\mathcal{L}_\epsilon}^{K} w_i(t)\right) & \text{, if } \mathcal{L}_\epsilon \leq K \leq \mathcal{R}_\epsilon \\ \frac{1}{W}\sum_{i=\mathcal{L}_\epsilon}^{\mathcal{R}_\epsilon} w_i(t)\overrightarrow{p(i)} & \text{, if } K > \mathcal{R}_\epsilon \end{cases}$$

the following inequality holds:

$$\boxed{\lVert\overrightarrow{\pi^*(t)} - \overrightarrow{\pi(t)}\rVert_v^\infty \leq \delta + \frac{3}{4}\varepsilon}$$

Here W, $w_i(t)$, $\mathcal{L}_\epsilon$, and $\mathcal{R}_\epsilon$ are computed using the Fox-Glynn algorithm, such that $\sum_{i=0}^{\mathcal{L}_\epsilon - 1}\gamma_i(t) \leq \frac{\varepsilon}{4}$, and $\sum_{i=\mathcal{R}_\epsilon+1}^{\infty}\gamma_i(t) \leq \frac{\varepsilon}{4}$.

Proof For proof details see [12]. □

Corollary 10 *Under the same conditions as Theorem 9:*

$$\boxed{\begin{aligned} &\lVert\overrightarrow{p^*} - \overrightarrow{p(K)}\rVert_v^\infty \leq \frac{\varepsilon}{4} \textit{ implies} \\ &\qquad \lVert\overrightarrow{\pi^*(t)} - \overrightarrow{\pi(t)}\rVert_v^\infty \leq \varepsilon \qquad (11) \end{aligned}}$$

When computing the probability *Prob*$(s,\ \mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G})$ we have $Ind = \mathcal{G}$ and $\overrightarrow{i_{Ind}} = \overrightarrow{i_\mathcal{G}}$. Recall that the main difference with the forward algorithm is that we now employ a global model-checking procedure, i.e., probabilities *Prob*$(s,\ \mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G})$ are determined for *all* states s. To guarantee an overall error bound of ε, one should use $\lVert\overrightarrow{p^*} - \overrightarrow{p(K)}\rVert_v^\infty \leq \frac{\varepsilon}{4}$ as a stopping criterion. Although our result at first sight looks quite similar to that in [24], there are various small, though important differences. As for the forward case, the influence of weights (that may yield an additional error) is taken into account. Secondly, the change of the summation lower bound from $i = 0$ to $i = \mathcal{L}_\epsilon$ in equation (8) is implicitly taken care of due to the fact that $\forall i < \mathcal{L}_\epsilon : w_i(t) = 0$. If weights are neglected, as in Theorem 6, an error bound is obtained that is too liberal. Finally, we remark that the steady-state detection error is one-sided for backward computations.

Table 1. The summary of results

Forward computations
Malhotra's result [16]: $\lVert\overrightarrow{p^*(0)} - \overrightarrow{p(0,K)}\rVert_v \leq \frac{\varepsilon}{4}$ implies $\lVert\overrightarrow{\pi^*(0,t)} - \overrightarrow{\pi(0,t)}\rVert_v \leq \varepsilon$ (7)
Our result: $\lVert\overrightarrow{p^*(0)} - \overrightarrow{p(0,K)}\rVert_v^\infty \leq \frac{\varepsilon}{8\lvert Ind\rvert}$ implies $\left\lvert\sum_{j\in Ind}\left(\pi^*(0,t)_j - \pi(0,t)_j\right)\right\rvert \leq \varepsilon$ (10)
Backward computations
Younes's result [24]: $\lVert\overrightarrow{p^*} - \overrightarrow{p(K)}\rVert_v \leq \frac{\varepsilon}{8}$ implies $\forall j \in S : -\frac{\varepsilon}{4} \leq \pi^*(t)_j - \pi(t)_j \leq \frac{3}{4}\varepsilon$ (9)
Our result: $\lVert\overrightarrow{p^*} - \overrightarrow{p(K)}\rVert_v^\infty \leq \frac{\varepsilon}{4}$ implies $\lVert\overrightarrow{\pi^*(t)} - \overrightarrow{\pi(t)}\rVert_v^\infty \leq \varepsilon$ (11)

5.3. Summary of results

To summarize we provide the Table 1. Please take into account that our results require the use of weights for the computations of $\overrightarrow{p^*(0)}$ and $\overrightarrow{p^*}$ as well as different left and right truncation points for the computation of Poisson probabilities (cf. Proposition 5).

6. Safely detecting stationarity

Although the (theoretical) results obtained so far in this paper provide safe criteria for detecting whether an equilibrium has been reached, they suffer from the problem that the stopping criterion refers to the steady-state vector $\overrightarrow{p^*}$ that is typically unknown. A possible way to circumvent this is to use the absolute convergence test (see Section 2.3 and 4.1). This boils down to comparing probability vectors that are $M > 0$ iterations apart. This, however, introduces an unknown error. To avoid this unpredictable error, we suggest to exploit the typical structure of the CTMC for time-bounded reachability. Recall that for checking the formula $\mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G}$, all states in $\mathcal{G}$ and in $\mathcal{I} = S \setminus (\mathcal{A} \cup \mathcal{G})$ are made absorbing [2]. Intuitively speaking, on the long run, the probability mass will flow to the states in $\mathcal{G}$ and in $\mathcal{I}$, or to bottom strongly connected components (BSCCs)—SCCs that once entered cannot be left anymore—in the remainder of the CTMC, i.e., in the set of states $S \setminus (\mathcal{G} \cup \mathcal{I})$. It can be shown (see below), that we can safely replace each of these BSCCs by a single absorbing state without affecting the validity of the time-bounded reachability problem. Checking for an equilibrium now amounts to check whether the residual probability mass in the remaining non-absorbing states is below a certain threshold. Let $B_{\mathcal{A},\mathcal{G}} = \{s \in B \cap (\mathcal{A} \setminus \mathcal{G})\,|B \text{ is a BSCC in } \mathcal{Q}[\mathcal{I} \cup \mathcal{G}]\}$.

Proposition 11 *For any state s in CTMC $(S,\ \mathcal{Q})$, time-bounded property $\mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G}$ and $\mathcal{Q}^B = \mathcal{Q}[\mathcal{I} \cup \mathcal{G}][B_{\mathcal{A},\mathcal{G}}]$*

we have:

$$\mathit{Prob}(s,\,\mathcal{A}\,\mathrm{U}^{[0,t]}\,\mathcal{G})\ \mathit{in}\ (S,\,\mathcal{Q})\ =\ \mathit{Prob}(s,\,S\,\mathrm{U}^{[t,t]}\,\mathcal{G})\ \mathit{in}\ (S,\mathcal{Q}^B)$$

Every state $s \in \mathcal{A} \setminus (\mathcal{G} \cup B_{\mathcal{A},\mathcal{G}}) = S \setminus (\mathcal{I} \cup \mathcal{G} \cup B_{\mathcal{A},\mathcal{G}})$ is a transient state. This follows directly from the construction of $\mathcal{Q}^B$.

Forward computations As the probability mass in transient states of $\mathcal{Q}^B$ on the long run equals 0, this can now be exploited. Due to the uniformization procedure the same is valid for the stochastic matrix $\mathcal{P}_B$ obtained after uniformizing CTMC $(S,\ \mathcal{Q}^B)$. When i increases, while computing $\overrightarrow{1_s} \cdot \mathcal{P}_B^i$, the probability to be in a transient state is only decreasing, and the probability to be in an absorbing state is increasing.

Theorem 12 *For the stochastic matrix $\mathcal{P}_B$ obtained after uniformizing CTMC $(S,\ \mathcal{Q}^B)$, for any K and $\delta > 0$ the following holds:*

$$\sum_{j \in \mathcal{A} \setminus \left(\mathcal{G} \cup B_{\mathcal{A},\mathcal{G}}\right)} p(s,K)_j \leq \delta \Rightarrow \forall i \geq K : \|\overrightarrow{p^*(s)} - \overrightarrow{p(s,i)}\|_v^\infty \leq \delta$$

Where $p(s,i)_j$ is the j'th component of $\overrightarrow{p(s,i)} = \overrightarrow{1_s} \cdot \mathcal{P}_B^i$, and $\overrightarrow{p^*(s)}$ is the steady-state probability for $\mathcal{P}_B$ when starting from state s.

Notice that this theorem gives a precise error bound for a steady-state detection. Still the convergence rate is not known so the check for steady-state should be performed every M iterations as before.

Backward computations The backward algorithm is based on $\overrightarrow{p(i)} = \mathcal{P}^i \cdot \overrightarrow{i_\mathcal{G}}$, where vector $\overrightarrow{i_\mathcal{G}}$ is not a distribution. The idea of backward computations is to accumulate the probability to reach states in $\mathcal{G}$. Information about the probability reaching $\mathcal{A} \setminus (\mathcal{G} \cup B_{\mathcal{A},\mathcal{G}})$ or $B_{\mathcal{A},\mathcal{G}} \cup \mathcal{I}$ is, however, not available. To compute the precise equilibrium, we propose to compute, in addition to $\overrightarrow{p(i)}$, the probability to be in either $B_{\mathcal{A},\mathcal{G}}$ or $\mathcal{I}$ after i steps.

Theorem 13 *For the stochastic matrix $\mathcal{P}_B$ obtained after uniformizing CTMC $(S,\ \mathcal{Q}^B)$, for any K and $\delta > 0$ the following holds:*

$$\|\overrightarrow{1} - \left(\overrightarrow{p(K)} + \overrightarrow{p^B\,(K)}\right)\|_v^\infty \leq \delta \Rightarrow \forall i \geq K : \|\overrightarrow{p^*} - \overrightarrow{p(i)}\|_v^\infty \leq \delta$$

where $\overrightarrow{p(i)} = \mathcal{P}_B^i \cdot \overrightarrow{i_\mathcal{G}}$, $\overrightarrow{p^B\,(i)} = \mathcal{P}_B^i \cdot \overrightarrow{i_{B_{\mathcal{A},\mathcal{G}} \cup \mathcal{I}}}$, and $\overrightarrow{p^*} = \lim_{i \to \infty} \mathcal{P}_B^i \cdot \overrightarrow{i_\mathcal{G}}$.

7. Experimental results

This section reports on some experiments that we conducted with existing and the proposed approaches towards on-the-fly steady-state detection. The experiments concentrate on illustrating the phenomenon of premature stationarity in existing model checkers for CTMCs. This is first shown by means of a simple, though artificial example. The fact that these phenomena occur in realistic examples too is illustrated by means of the workstation cluster [7, 4, 23, 14, 18], an example that has established itself as one of the benchmarks for probabilistic model checking. (A similar phenomenon has been reported for a group communication protocol in [17].) We finally report on the computation time needed for our proposed algorithms. The tools that are used in the experiments are *Prism v2.1* [15], *ETMCC v1.4.2* [9] and our model checker called *MRMC v1.0* [10]. The first two support an on-the-fly steady-state detection as described in Section 4.1, whereas MRMC realizes (as an option) the algorithms proposed in this paper. GreatSPN v1.0 [5] uses ETMCC as a back-end and the results reported on ETMCC therefore also apply to GreatSPN.

All experiments are conducted on a *Pentium 4 3.00GHz, 2Gb RAM, Suse Linux* machine, and are focused on the backward algorithm. For comparison reasons, exact probabilities obtained from Matlab are used. For presented examples, curves obtained with Matlab and MRMC, with the steady-state detection turned on, coincide.

It should be noted that each tool uses different M for steady-state detection, when employing relative and absolute convergence tests. For example, PRISM uses $M = 1$, which allows to save on memory usage, while ETMCC uses $M = 10$. This results in the fact that PRISM detects a steady-state earlier than ETMCC.

A slowly convergent CTMC. Consider the CTMC in Fig. 1 and let $\mathcal{A} = \{0, 1\}$ and $\mathcal{G} = \{2\}$. The peculiarity of this CTMC is that the probability to move from the starting state 1 to the goal state is very low. Fig. 2 plots the probability $\mathit{Prob}(1,\ \mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G})$ for the tools considered. The experiments for PRISM are performed using either a relative (rel) or an absolute (abs) convergence test. As we want to make these two convergence tests behave similarly, the relative error is set to 10^{-1}. This approximately corresponds to an absolute error 10^{-6}. Note that ETMCC, and both variants of PRISM (abs and rel) detect stationarity prematurely whereas MRMC does not. For the indicated range of t, the resulting error is within the inaccuracy $\varepsilon = 10^{-6}$; for larger values of t (upto around 16,000), the resulting probabilities for ETMCC and PRISM differ more than ε (and MRMC still does not detect an equilibrium). More details on the iteration index at which an equilibrium is detected (K), and the corresponding probability are given in Table 2.

To validate the tweak of the relative error bound for PRISM, it should be noted that with the original error bound 10^{-6}, the premature steady-state detection still occurs but for larger values of the time bound t, such as $t \geq 1050000$.

Table 2. Steady-state detected on iteration K

Tool	Error	K	$\mathcal{P}^K \cdot \overrightarrow{i_\Psi}$
Prism v2.1(abs)	10^{-6}	2	$(5.00025 \cdot 10^{-5}, 2.5 \cdot 10^{-9}, 1.0)$
Prism v2.1(rel)	10^{-1}	12	$(5.00275 \cdot 10^{-5}, 2.75 \cdot 10^{-8}, 1.0)$
ETMCC v1.4.2	10^{-6}	20	$(5.00475 \cdot 10^{-5}, 4.75 \cdot 10^{-8}, 1.0)$
MRMC v1.0	10^{-6}	—	—

Workstation cluster. As a larger and more realistic example, we considered the workstation cluster as originally proposed in [7]. This example is used as a benchmark in various papers, e.g., [4, 23, 14, 18]. The example consists of two symmetric subsystems both consisting of L workstations. Inside a subsystem, the workstations are connected by means of switches that are connected by a backbone. Each component of the system (workstation, switch and backbone) is failure prone. Depending on the number of operational and connected workstations, the system is said to offer maximum or minimum quality-of-service. The time-bounded reachability property considered is the probability to eventually reach a service level below the minimum. The investigated configuration is L=5, and the minimum QoS equals 3. The resulting CTMC has about 5,000 states.The rates of the model are taken from the PRISM web page. Fig. 3 plots the computed probabilities using PRISM and ETMCC using the absolute error 10^{-6} and relative error 10^{-3}. Where $\mathcal{A} =$ *all states* and $\mathcal{G} =$ *states for which not minimum holds*.The effect is the same as for the artificial example shown before. Note that with the default relative error 10^{-6}, PRISM prematurely detects steady state with the values of time $t \geq 28000$.

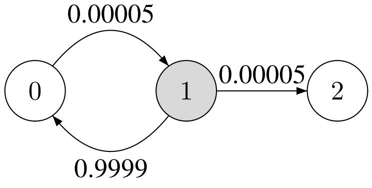

Figure 1. A slowly convergent CTMC

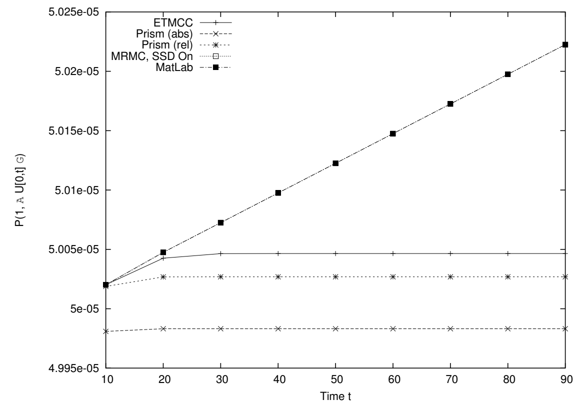

Figure 2. *Prob*$(1,\ \mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G})$ for various t

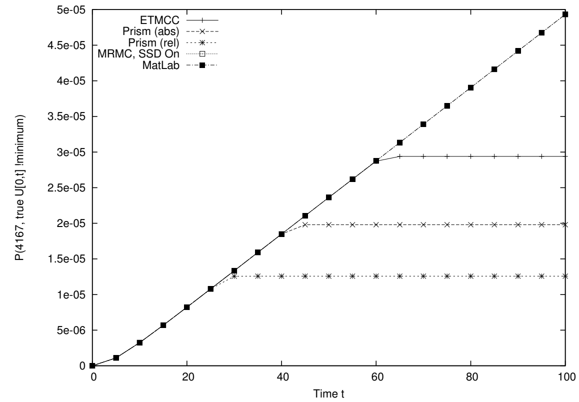

Figure 3. *Prob*$(4167,\ \mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G})$ for various t

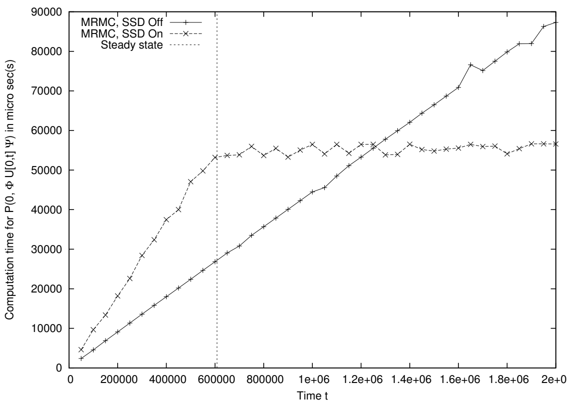

Figure 4. Time required for computing *Prob*$(0,\ \mathcal{A}\ \mathrm{U}^{[0,t]}\ \mathcal{G})$ with and without steady-state detection

Runtime. To conclude, we investigated the impact of the proposed on-the-fly steady-state detection algorithm on the verification time (for backward computations). The typical pattern obtained is depicted in Fig. 4. Prior to the point from which a steady state is detected during computations, the run time is doubled. This is due to the fact that for the backward algorithm $\overrightarrow{p^B(i)}$ is computed in addition to $\overrightarrow{p(i)}$. Once the equilibrium is reached (and detected), the run time for the variant with steady-state detection remains constant, whereas the runtime of the plain algorithm continues to grow linearly in t.

8. Concluding remarks

This paper presented refined error bounds for existing standard transient analysis and for time-bounded reachability algorithms that incorporate on-the-fly steady-state detection. These results are obtained using a refined bound for the Fox-Glynn algorithm. (Proofs are omitted in this paper; all of them can be found in [12].) These results are complemented by a simple technique to *safely* detect a steady-state for standard transient analysis and time-bounded reachability. Experiments showed the impact of this new algorithm. Our backward algorithm increases runtime (factor two), and requires two extra vectors. For the forward algorithm there is no increase of time, and no additional space is required. In both cases it is guaranteed to avoid detecting premature equilibria.

Acknowledgment The authors thank *David N. Jansen* for his careful and detailed comments and for pointing out a flaw in a previous version of the paper. *Dave Parker* is thanked for discussions about steady-state detection in PRISM.

This research has been performed as part of the MC=MC project that is financed by the Netherlands Organization for Scientific Research (NWO).

References

[1] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Model-checking continuous-time Markov chains. *ACM Transactions on Computational Logic*, 1(1):162–170, 2000.

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

[3] P. Buchholz, M. Fischer, P. Kemper, and C. Tepper. Model checking of CTMCs and discrete event simulation integrated in the APNN-Toolbox. In *Measurement, Modelling, and Evaluation of Computer-Communication Systems*, volume 781, pages 30–33, 2003.

[4] P. Buchholz, J.-P. Katoen, P. Kemper, and C. Tepper. Model-checking large structured Markov chains. *Journal of Logic and Algebraic Programming*, 56:69–96, 2003.

[5] D. D'Aprile, S. Donatelli, and J. Sproston. CSL model checking for the GreatSPN Tool. In *Int. Symp. on Computer and Information Sciences*, volume 3280 of *LNCS*, pages 543–552, 2004.

[6] B. L. Fox and P. W. Glynn. Computing Poisson probabilities. *Commun. ACM*, 31(4):440–445, 1988.

[7] B. Haverkort, H. Hermanns, and J.-P. Katoen. On the use of model checking techniques for dependability evaluation. In *Reliable Distributed Systems (SRDS'00)*, pages 228–237, 2000.

[8] B. R. Haverkort. *Performance of Computer Communication Systems: A Model-Based Approach*. New York, NY, USA, 1998.

[9] H. Hermanns, J.-P. Katoen, J. Meyer-Kayser, and M. Siegle. A tool for model-checking Markov chains. *Int. J. on Softw. for Technology Transfer (STTT)*, 4(2):153–172, 2003.

[10] J.-P. Katoen, M. Khattri, and I. S. Zapreev. A Markov reward model checker. In *Quantitative Evaluation of Systems (QEST)*, pages 243–244, 2005.

[11] J.-P. Katoen, M. Kwiatkowska, G. Norman, and D. Parker. Faster and symbolic CTMC model checking. In *Process Algebra and Probabilistic Methods, Performance Modeling and Verification (PAPM/PROBMIV'01)*, volume 2165 of *LNCS*, pages 23–38, 2001.

[12] J.-P. Katoen and I. S. Zapreev. Safe on-the-fly steady-state detection for time-bounded reachability. Technical Report TR-CTIT-05-52, CTIT, University of Twente, 2005.

[13] M. Kwiatkowska. Model checking for probability and time: From theory to practice. In *Logic in Computer Science (LICS)*, pages 351–360, 2003.

[14] M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic symbolic model checker. In *Modelling Techniques and Tools for Computer Performance Evaluation (TOOLS'02)*, volume 2324 of *LNCS*, pages 200–204, 2002.

[15] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 2.0: A tool for probabilistic model checking. In *Quantitative Evaluation of Systems (QEST)*, pages 322–323, 2004.

[16] M. Malhotra, J. K. Muppala, and K. S. Trivedi. Stiffness-tolerant methods for transient analysis of stiff Markov chains. *Microelectronics and Reliability*, 34(11):1825–1841, 1994.

[17] M. Massink, J.-P. Katoen, and D. Latella. Model checking dependability attributes of wireless group communication. In *Dependable Systems and Networks (DSN)*, pages 711–720, 2004.

[18] Prism web-page, Workstation cluster example.
http://www.cs.bham.ac.uk/ dxp/prism/casestudies/cluster.php.

[19] K. Sen, M. Viswanathan, and G. Agha. On statistical model checking of stochastic systems. In *Computer Aided Verification (CAV)*, volume 3576 of *LNCS*, pages 266–280, 2005.

[20] W. J. Stewart. A comparison of numerical techniques in Markov modeling. *Commun. ACM*, 21(2):144–152, 1978.

[21] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

[22] H. C. Tijms. *A First Course in Stochastic Models*. John Wiley & Sons, Ltd, 2003.

[23] H. Younes, M. Kwiatkowska, G. Norman, and D. Parker. Numerical vs. statistical probabilistic model checking: An empirical study. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 2988 of *LNCS*, pages 46–60, 2004.

[24] H. Younes, M. Kwiatkowska, G. Norman, and D. Parker. Numerical vs. statistical probabilistic model checking. *Int. J. on Softw. for Technology Transfer (STTT)*, 2005.

[25] H. L. S. Younes. Ymer: A statistical model checker. In *Computer Aided Verification (CAV)*, volume 3576 of *LNCS*, pages 429–433, 2005.