Continuous-Time Markov Chain with Alarms analysis and comparaison of tools *

Thomas Mari[1[0000−1111−2222−3333]], Second Author[2,3[1111−2222−3333−4444]], and Third Author[3[2222−−3333−4444−5555]]

[1] ENS-Rennes, Campus de Ker lann, Avenue Robert Schuman, 35170 Bruz, France
[2] Masaryk University, Botanick 68A, 602 00 Brno-Krlovo Pole, Czech republic
lncs@springer.com
http://www.springer.com/gp/computer-science/lncs
[3] ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
{abc,lncs}@uni-heidelberg.de

Abstract. Phase-type fitting remains the only way of modeling non-Markovian distributions within PRISM model checker. When analyzing models with phase-type distributions, it is problematic to obtain results of sufficient precision within reasonable time. We ran experimental computations and deduced a reliable way of obtaining precise analysis results for phase-type fitted PRISM CTMCs. Lastly, we hint at an entirely different (and arguably better) approach to handling non-Markovian distributions.

Keywords: PRISM model checker · CTMC · deterministic timeout · phase-type distribution · modeling · analysis.

1 Introduction

PRISM [1] and Storm [2] are popular tools for modeling and analysis of stochastic systems in continuous time. They use efficient algorithms for analysis of continuous time Markov chains (CTMC). This approach suffers from a severe restriction of expressiveness, because the time between transitions must be exponentially distributed. This restriction can be remedied by the use of phase-type distributions, which can approximate any general distribution with arbitrary accuracy by only using exponential distributions [3]. However, the use of phase-type distributions drastically increases the number of states within the CTMC.

In this paper, we experimentally evaluate the precision of the result and required computation time of various approaches to analysis of continuous time stochastic models with deterministic transitions (timeouts). The obtained results are then compared against results of our extension of PRISM capable of efficiently analyzing CTMC with non-Markovian alarms (ACTMC).

We will go through the theoretical bases needed to understand the contribution. CITE LUBOSH'THESIS

* Supported by organization x.

2 Related Work

There are various tools and models in the area of probabilistic verification, especially in the analysis of non-Markovian Model. One of those model is stochastic Petri Nets. Oris is a tool for the analysis of stochastic Petri net [4]. Oris allows to analyze Petri nets with plenty of time distributions for each event such as exponential, Dirac, Uniform or instant. The expressiveness is sufficient to work on Petri nets that are equivalent to an ACTMC. This tool compute the Steady states probability as well as well as the transient state probability which is state probability at a given time.

A less expressive model is the CTMC defined in subsection 3.2, PRISM[1] and Storm [2] are both well known tool to analyze CTMC. The engine of those tools are quite efficient for analysis due to the memoryless property of the exponential distribution. One can use phase-type (PH) approximation to approach a non-Markovian model. The downside of PH approximation is the increase of the number of states which can slow down the process of annalysis. PH Fitting can be challenging depending on the PH distribution. There is automatic tools that do the Phase-Type Fitting like HyperStar [5]. The power of this tools is to work on real dataset. According the system you want to model is it indeed practical to use real data such that the model depict more precisely the system.

An alternative of PH-approximation is to work directly with a d-CTMC which is a CTMC plus deterministic distribution. This paper [6] depict an efficient approach to analyze ACTMC with distribu- tions where phase-type increase significantly the number of states with reasonable precision. Those distributions are amongst shifted exponential distribution and fixed-delay distribution.

A different objective on the analysis of non-Markovian model is to synthesize optimal parameter. We can add a reward system on the model and computed the optimal parameter, like a timeout on some transitions such that the mean-payoff is ϵ-optimal [6]. The mean-payoff can be seen as the average long term efficiency of the system.

3 Preliminaries

3.1 Time distribution

The class of model we use is probabilistic continuous time model, which means that the time passed in a state will depend on a time distribution. A time distribution can be depicted by its cumulative distribution function (CDF) $F : \mathbb{R} \to [0, 1]$, $F(t)$ is the probability that the event will happend before the time t. The exponential distribution of rate λ has a CDF $F(t) = 1 - \exp(-\lambda t)$, it is important to keep in mind that a rate refers to the exponential distribution. The Dirac distribution of timeout τ has a CDF $F(t) = \chi_{[\tau,+\infty[}(t) = \begin{cases} 0 \text{ if } t < \tau \\ 1 \text{ else} \end{cases}$. In this paper, time distribution which are not exponential will be called non-Markovian distribution. Hence, a non-Markovian model will refer to a model with non-Markovian distribution.

3.2 CTMC

A continuous-time Markov chain (CTMC) is a triple $C = (S, Q, s_{in})$, where

- S is a finite set of states
- $Q : S \times S \to \mathbb{R}_{\geq 0}$ is a matrix of rates such that $\sum_{s' \in S} Q(s, s') > 0$ for each $s \in S$
- $s_{in} \in S$ is an initial state

$Q(s, s')$ is the rate of the transition from s to s'. We define the exit rate of state s as $\lambda_s = \sum_{s' \in S} Q(s, s')$

A run of a CTMC $\mathcal{C}$ is an infinite alternating sequence of states and times $\omega = s_0 t_0 s_1 t_1 ...$ where

- $s_0 = s_{in}$
- s_i is the i-th state
- t_i is the time spent in s_i

For each $i \in \mathbb{N}$

- t_i is choosen randomly according to the exponential distribution with rate λ_{s_i}
- s_{i+1} is choosen according to the discrete distribution $\dfrac{Q(s_i, .)}{\lambda_{s_i}}$

Without considering the time spent on each state, CTMC works the same way than discrete time Markov Chain. It is interesting to understand why it is difficult to allow non-Markovian distribution. In this case, when available transitions are not exponential, we cannot pick the next state as easily. We will see an extension of CTMC, where we allow at most one non-Markovian at the same time.

3.3 ACTMC

We can see ACTMC as CTMC with some events called alarms that are enabled in disjoints sets of states and those events has their own timer which is active only when inside the enabled set and happend according to some continuous time distribution. Those alarms works concurrently with the exponential transition seen in the CTMC.

$\mathcal{A} = (S, Q, s_{in}, A, \langle S_a \rangle, \langle P_a \rangle, \langle F_a \rangle)$ where :

- (S, Q, s_{in}) is a CTMC
- A is a set of alarms
- $\langle S_a \rangle = (S_a)_{a \in A}$, the set of states where a is enabled
  - if $a \neq a'$ then $S_a \bigcap S_{a'} = \emptyset$
- $\langle P_a \rangle = (P_a)_{a \in A}$ where P_a is a probability matrix
- $\langle F_a \rangle = (F_a)_{a \in A}$ where F_a is a cumulative distribution function (CDF)

Operational Behavior

A run of a ACTMC $\mathcal{A}$ is an infinite alternating sequence of states and times $(s_0, \eta_0)t_0(s_1, \eta_1)t_1...$ where

- $s_0 = s_{in}$
- s_i is the i-th state
- t_i is the delay in s_i, t_i is the time spent in s_i without alarms
- η_i is the value of the timer, the remaining time for the alarm to ring.

We define $S_{off} = S \setminus \bigcup_{a \in A} S_a$ the set of states where no alarms are enabled

For initialization

- $s_0 = s_{in}$
- $\eta_0 = \begin{cases} \infty \text{ if } s_0 \in S_{off} \\ \text{randomly choosen according to } F_a \text{ if } s_0 \in S_a \end{cases}$

For each $i \in \mathbb{N}, t_i$ is choosen randomly according to the exponential distribution λ_{s_i}

Two cases are possible, either the alarm ring ($\eta_i \leq t_i$) or t_i is too short and the alarm doesn't ring.

If $\eta_i \leq t_i$, the alarm ring

- $t_i := \eta_i$ the value of delay is overwritten and match the time spent in s_i
- s_{i+1} is choosen according to the discrete distribution $P_a(s_i, .)$
- $\eta_{i+1} = \begin{cases} \infty \text{ if } s_{i+1} \in S_{off} \\ \text{randomly choosen according to } F_a \text{ if } s_{i+1} \in S_a \end{cases}$

If $\eta_i > t_i$, the alarm does not ring

- t_i the value of delay remain the same and match the time spent in s_i
- s_{i+1} is choosen according to the discrete distribution $\dfrac{Q(s_i, .)}{\lambda_{s_i}}$
- $\eta_{i+1} = \begin{cases} \infty \text{ if } s_{i+1} \in S_{off} \\ \eta_i - t \text{ if } s_{i+1} \in S_a \text{ and } s_i \in S_a \\ \text{randomly choosen according to } F_a \text{ if } s_{i+1} \in S_a \text{ and } s_i \notin S_a \end{cases}$

If the state remain in the set enabled by the alarm, it is updated. Otherwise it is reset according to the new enabled set.

3.4 Steady State Probability

In a CTMC, it is important to understand the long term behavior of the system. If the CTMC modelize some production, we want to know the ratio of time it will spend in critical state or the expected efficiency of the system. It is possible to parametrized a CTMC with a reward system. For each state, we affect a reward that is a real number. During a run, we can computed the payoff of the run by adding the reward of each state multiplied by the time spend in the state. To compute a mean payoff we can use the Steady state Probability (SSP) defined as below. let $p(t)$ be the expected distribution at the time t. $p_i(t)$ is the i-th component of the distribution, i.e. the probability of the i-th state at the time t. $p = (p_i)_{i=1}^{|S|} = (\lim_{t\to\infty} p_i(t))_{i=1}^{|S|}$

We will compare the different tools, models and methods considering only the the SSP computation.

3.5 Phase Type fitting

Phase Type fitting (PH fitting) is approaching a non-Markovian distribution by a sequence of exponential distributions. Consider an infinite sequence of finite sequences of rates of length k $((\lambda_{i,k})_{i=0}^{k})_{k\in\mathbb{N}}$. We want that the distribution of the sequence of exponential transitions to converge to the non-Markovian distribution.

To obtain analysis on ACTMC, we use PH fitting to create CTMC from ACTMC, and analyse those CTMC Then we deduce results on the ACTMC. The reason is that Prism or Storm can use CTMC and not ACTMC. Figure 1 and figure 2 are a toy example of the PH fitting of a simple ACTMC $\mathcal{A} = (\{A, B\}, \begin{bmatrix} 0.5 & 0 \\ 2 & 1 \end{bmatrix}, A, \{A\}, \begin{bmatrix} 1 & 0 \end{bmatrix}, \{d\})$ with constant parameter k. The created CTMC is $\mathcal{C} = (\{A, B, 1, 2, ..., k-1, B\}, \begin{bmatrix} 0 & \lambda_{1,k} & 0 & 0 & ... & 0 \\ 0 & 0 & \lambda_{2,k} & 0 & ... & 0 \\ 0 & 0 & 0 & \lambda_{3,k} & ... & 0 \\ 0 & 0 & 0 & 0 & ... & 0 \\ ... & ... & ... & ... & ... & \lambda_{k,k} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, A)$

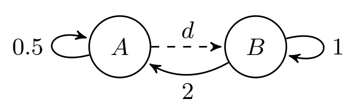

Fig. 1. Model of a simple ACTMC $\mathcal{A}$, d is a non-Markovian distribution.

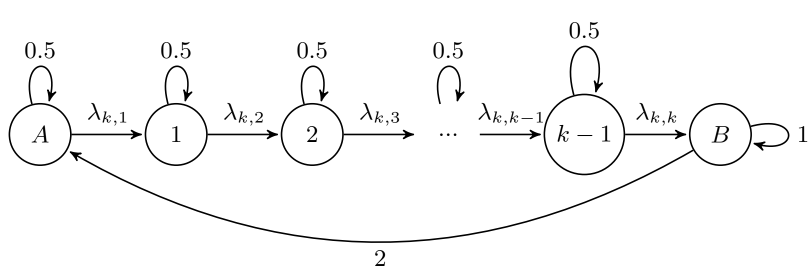

Fig. 2. Model of a CTMC obtain by PH fitting of the ACTMC figure 1.

In order to deduce the SSP of Figure 1, we will add the SSP of Figure 2. The sum of the states $\{A, 1, 2, ..., k-1\}$ will be the probability of the state A in the ACTMC. The probability of the state B in the ACTMC will be the probability of the state B in the CTMC.

4 Experimental evaluation

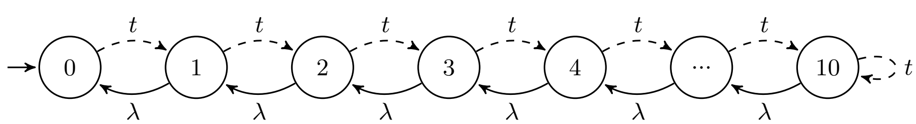

Fig. 3. Model of a D/M/1/10 queue with production timeout $t = 0.1$ and service rate $\lambda = \frac{ln(2)}{t} \approx 6.931471805599453094$. The deterministic arrival event does not reset when the exponential service event occurs.

We have chosen to model a D/M/1/10 queue (deterministic arrivals, exponentially distributed service, single server, capacity of 10) as shown in Figure 3. The deterministic arrivals are approximated by a phase-type distribution, so the model remains a CTMC and can be analyzed by PRISM. The PRISM model we used is shown in Figure 4. We computed the steady-state distribution with various values of PRISM termination epsilon ϵ and the number of phases k.

```
ctmc

const int k;
const int qCapacity = 10;
const double timeout = 0.1;
const double lambda = 6.931471805599453094;

module main

qSize : [0..qCapacity] init 0;

[produce] (qSize <= qCapacity) -> (qSize' = min(qSize+1,qCapacity));
[consume] (qSize > 0) -> lambda: (qSize' = qSize - 1);

endmodule

module trigger

i : [1..k+1];

[] i < k -> k/timeout : (i'=i+1);
[produce] i = k -> k/timeout : (i'=1);

endmodule
```

Fig. 4. PRISM CTMC model of the D/M/1/10 queue as shown in Figure 3 with a phase-type distribution approximating the arrival timeout. The phase-type module *trigger* is used as suggested on the PRISM website. Phase-type parameter k represents the number of phases. Increasing k should improve the approximation of the timeout.

4.1 Obtaining precise results for reference

In order to compare the precision of the results obtained by the experiments, a trustworthy high-precision reference is needed. To achieve this, we have modeled the given D/M/1/10 queue in Oris [4]. The steady-state distribution obtained by Oris equals the steady-state distribution obtained by our PRISM ACTMC implementation up to at least 14 digits. Hence, our tool can be considered a trustworthy reference with precision of 10^{-14}.

I THINK AT THIS POINT OF THE ARTICLE, IT IS NOT OBVIOUS THAT OUR TOOL REFER TO PRISM GSMP

4.2 Default absolute termination epsilon $\epsilon = 10^{-6}$

First, we used the PRISM default absolute termination epsilon $\epsilon = 10^{-6}$. The results are shown in Figure 5. For this ϵ, the obtained results are arguably insufficient. Increasing k does little to improve the precision, and only up to about $k \leq 50$, at which point increasing k further starts making the precision worse.

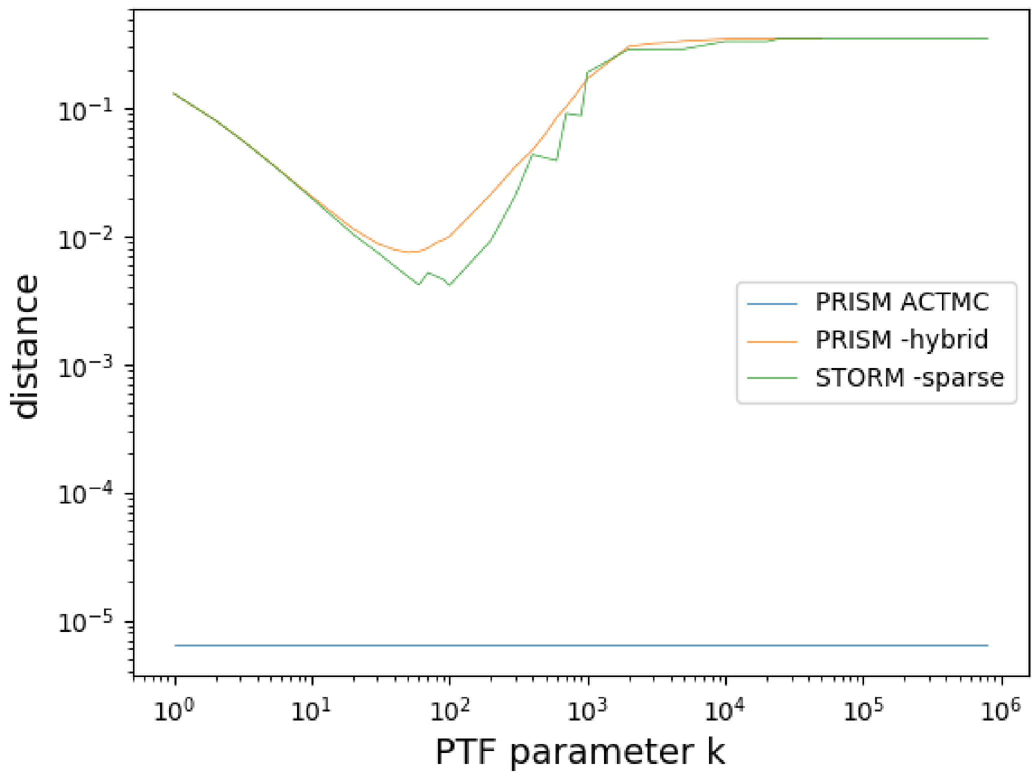

Fig. 5. Maximum distance (infinity norm) of the steady-state probabilities from the reference high-precision ACTMC (10^{-20}) against various values of k for default absolute termination epsilon $\epsilon = 10^{-6}$. ACTMC results for $\epsilon = 10^{-6}$ are also compared to the reference.

4.3 Lowering absolute termination epsilon to $\epsilon = 10^{-10}$

For much lower $\epsilon = 10^{-10}$, the precision of the result improves. However, for higher k the precision continues to deteriorate like in Section 4.3. The results are shown in Figure 6.

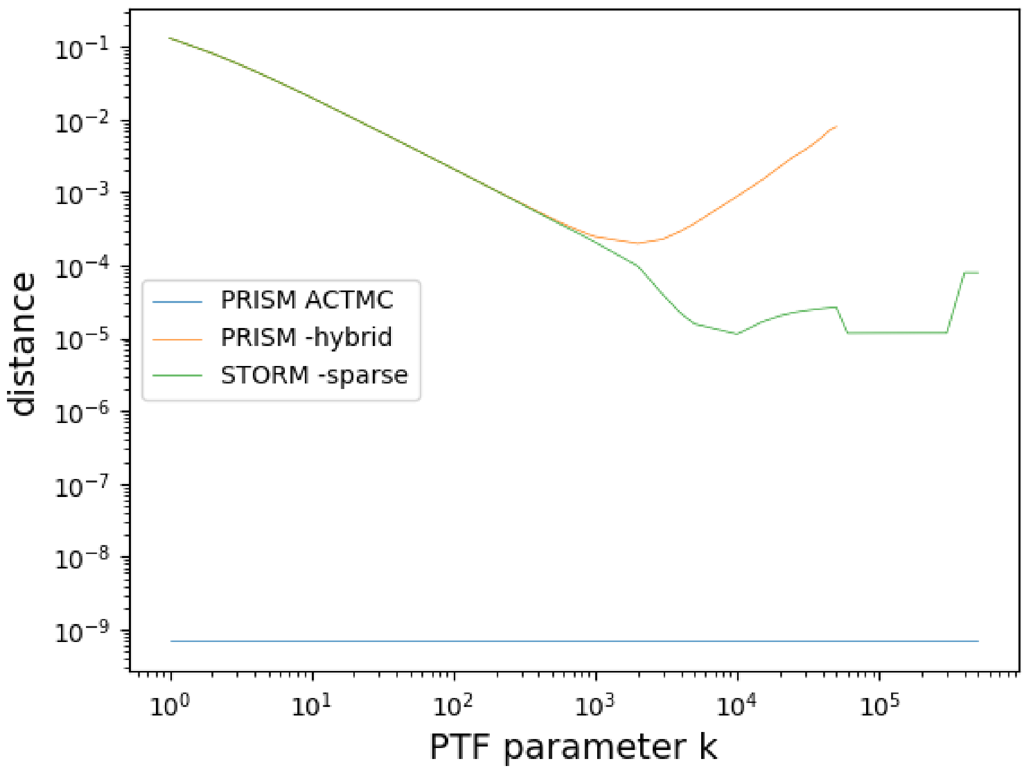

Fig. 6. Maximum distance (infinity norm) of the steady-state probabilities from the reference high-precision ACTMC (10^{-20}) against various values of k for lowered absolute termination epsilon $\epsilon = 10^{-10}$. ACTMC results for $\epsilon = 10^{-10}$ are also compared to the reference.

4.4 Adjusting epsilon $\epsilon = 10^{-10}$ to k

Since increasing k increases the amount of phases (intermediate states) and each phase may have error up to ϵ, the actual potential error for each state of the queue is $\epsilon \cdot k$. To compensate for this, we devise a simple formula

$$\epsilon^{'} = \frac{\epsilon}{k}$$

where $\epsilon^{'}$ is the adjusted epsilon that should be given to PRISM if precision ϵ is desired for k phases. Using adjusted epsilon, the results get significantly better, as shown in Figure 7. What's more, the phase-type fitting results now seem to converge towards the reference steadily as k increases. This indicates that this approach is correct, and that the reference is trustworthy.

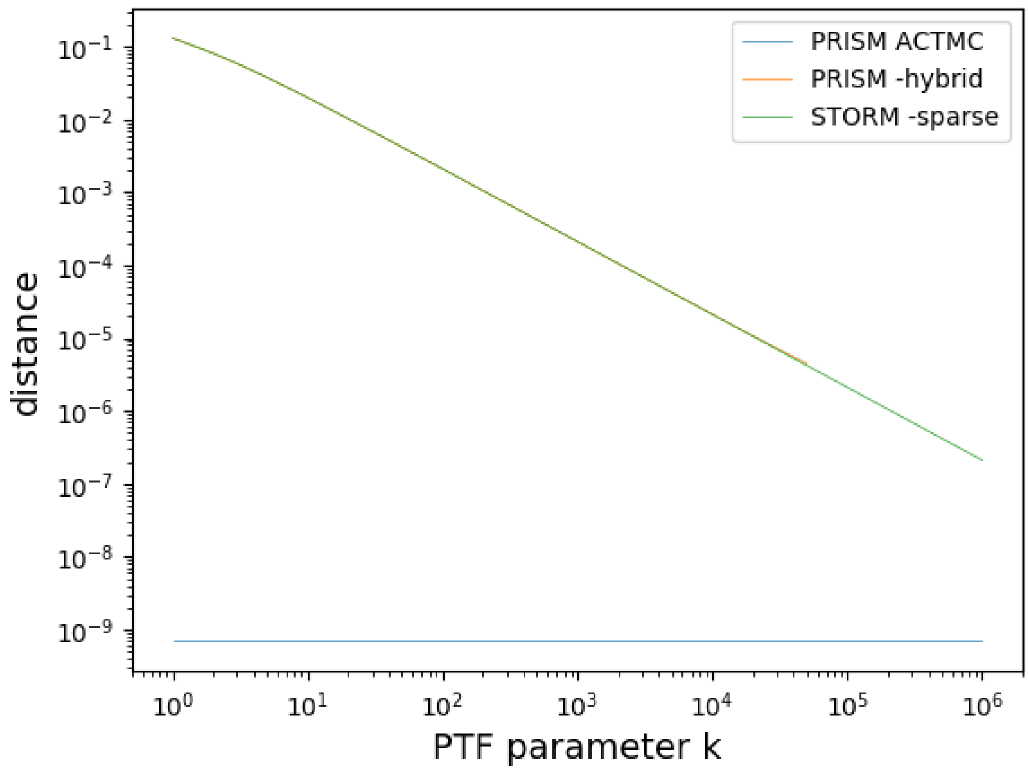

Fig. 7. Maximum distance (infinity norm) of the steady-state probabilities from the reference high-precision ACTMC (10^{-20}) against various values of k for lowered and adjusted absolute termination epsilon $\epsilon = 10^{-10}$. ACTMC results for $\epsilon = 10^{-10}$ are also compared to the reference.

4.5 Required computation time to obtain precise results

Although we have shown it is possible to obtain relatively precise results using phase-type distributions, the precision comes at a rather steep cost in computation time. This is shown in Figure 8. Note that even though Storm is one of the fastest available tools, it still struggles to deliver good precision within reasonable time. The computations all ran on the same virtual machine under same conditions[4].

4.6 Another Model : firstRejuvenation

We can observe similar results in terms of lack of precision on another model Figure 9. The precision must also depend on the PH fitting parameter k in this

[4] Running in a virtual machine might have reduced the performance. The virtual machine we used is available at the Storm website.

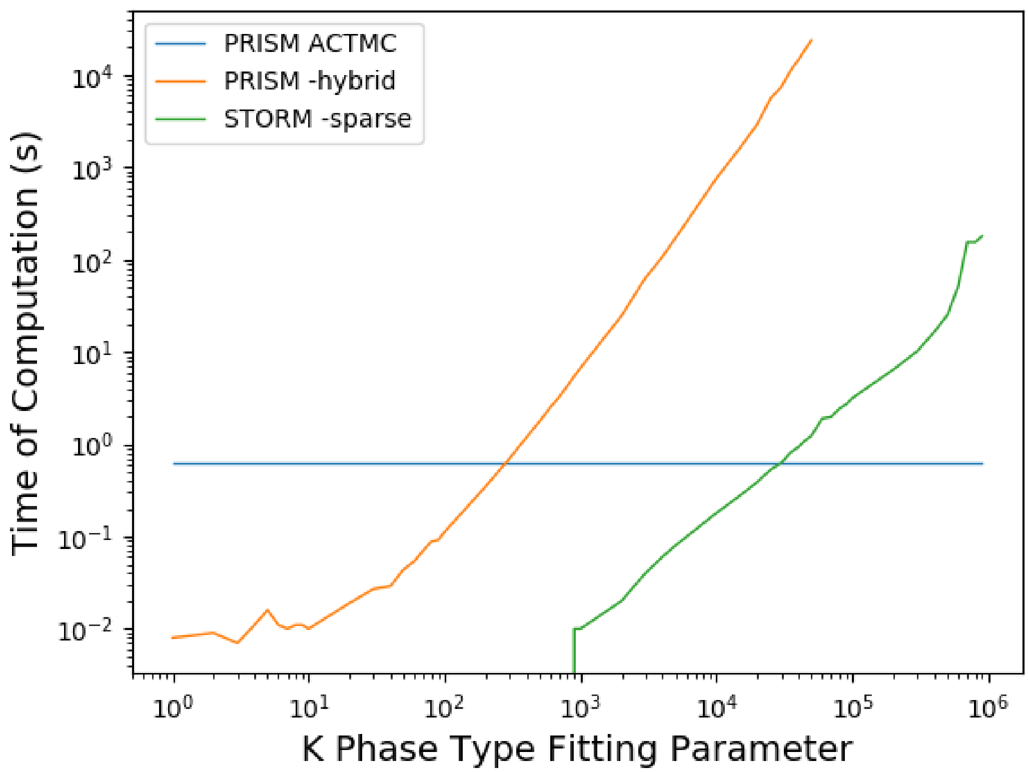

Fig. 8. Time of computation in seconds of the steady-state probabilities against various values of k for lowered and adjusted absolute termination epsilon $\epsilon = 10^{-10}$, i.e. corresponding to the data shown in Figure 7. The computation of our PRISM ACTMC is constant in regards to k because it does not use phase-type fitting.

model. Here we went through difficulties related to the methods used by the engine in Prism. The Power method for computing SSP uses different iterations which are matrix multiplication. The number of iteration will strongly depend on the model and its parameter like timeouts and rates. In the First Rejuvenation model The convergence for the power method is really slow and the number of iteration required is important. In the queue, 100 000 000 iterations was enough to compute the SSP, but here the number of iteration prevent us to compute the SSP for k greater than 1000 with reasonable precision.

The results of computation are given in the Figure 9. The orange curve is the error versus the number of phase k. The precision setting of the computation is 10^{-10} but the error is much bigger (at least 10^{-3}). This error can be related to the theoretical approximation error of the PH fitting, but also the error of computation of the engine.

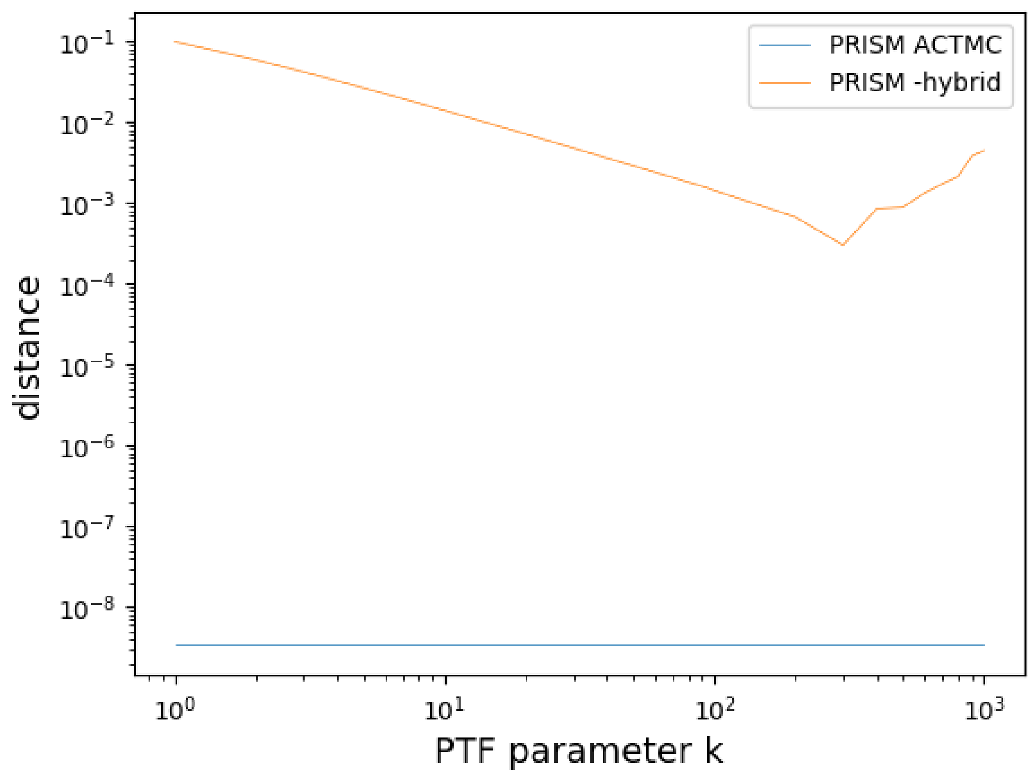

Fig. 9. Maximum distance (infinity norm) of the steady-state probabilities from the reference high-precision ACTMC (10^{-20}) against various values of k for lowered and adjusted absolute termination epsilon $\epsilon = 10^{-10}$. ACTMC results for $\epsilon = 10^{-10}$ are also compared to the reference.

5 Conclusion

We have presented a reliable method for high-precision analysis of PRISM CTMC models with phase-type distributions, that is, adjusting the termination epsilon by the number of phases. However, the large number of phases drastically increases the computation time.

Using the default setting of precision when doing PH fitting is not reliable at all. The relation between the experimental precision and the parameter of the model is not trivial. We have presented a sufficient condition to increased the precision while increasing the number of phases.

For better performance, we suggest usage of more specialized tools that can deal with non-Markovian distributions directly, without phase-type fitting. Our PRISM ACTMC extension [5] has delivered results with sufficient precision a lot faster.

References

[1] M. Kwiatkowska, G. Norman, and D. Parker. "PRISM 4.0: Verification of Probabilistic Real-time Systems". In: *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*. Ed. by G. Gopalakrishnan and S. Qadeer. Vol. 6806. LNCS. Springer, 2011, pp. 585–591.

[2] Christian Dehnert et al. "A storm is Coming: A Modern Probabilistic Model Checker". In: *CoRR* abs/1702.04311 (2017). arXiv: `1702.04311`. URL: `http://arxiv.org/abs/1702.04311`.

[3] Peter Buchholz, Jan Kriege, and Iryna Felko. *Input Modeling with Phase-Type Distributions and Markov Models: Theory and Applications*. Springer Publishing Company, Incorporated, 2014. ISBN: 3319066730, 9783319066738.

[4] Giacomo Bucci et al. "Oris: a tool for modeling, verification and evaluation of real-time systems". In: *International Journal on Software Tools for Technology Transfer* 12.5 (Sept. 2010), pp. 391–403. ISSN: 1433-2787. DOI: `10.1007/s10009-010-0156-8`. URL: `https://doi.org/10.1007/s10009-010-0156-8`.

[5] Philipp Reinecke, Tilman Krauß, and Katinka Wolter. "Phase-Type Fitting Using HyperStar". In: *Computer Performance Engineering*. Ed. by Maria Simonetta Balsamo, William J. Knottenbelt, and Andrea Marin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 164–175. ISBN: 978-3-642-40725-3.

[6] Christel Baier et al. "Mean-Payoff Optimization in Continuous-Time Markov Chains with Parametric Alarms". In: *CoRR* abs/1706.06486 (2017). arXiv: `1706.06486`. URL: `http://arxiv.org/abs/1706.06486`.

[5] Our PRISM ACTMC extension is not fully completed yet and lacks optimization. It will be officially released later, but early development version is available at https://github.com/VojtechRehak/prism-gsmp.