Adaptive Formal Approximations of Markov Chains

Alessandro Abate[a], Roman Andriushchenko[b], Milan Češka[b,*], Marta Kwiatkowska[a]

[a] *Department of Computer Science, University of Oxford, Oxford, UK*
[b] *Faculty of Information Technology, IT4Innovations Centre of Excellence, Brno University of Technology, Brno, Czech Republic*

Abstract

We explore formal approximation techniques for Markov chains based on state-space reduction that aim at improving the scalability of the analysis, while providing formal bounds on the approximation error. We first present a comprehensive survey of existing state-reduction techniques based on clustering or truncation. Then, we extend existing frameworks for aggregation-based analysis of Markov chains by allowing them to handle chains with an arbitrary structure of the underlying state space – including continuous-time models – and improve upon existing bounds on the approximation error. Finally, we introduce a new hybrid scheme that utilises both aggregation and truncation of the state space and provides the best available approach for approximating continuous-time models. We conclude with a broad and detailed comparative evaluation of existing and new approximation techniques and investigate how different methods handle various Markov models. The results also show that the introduced hybrid scheme significantly outperforms existing approaches and provides a speedup of the analysis up to a factor of 30 with a corresponding approximation error bounded within 0.1%.

Keywords: Markov models, probabilistic model checking, approximation techniques, adaptive aggregation

1. Introduction

Probability plays a prominent role in the design and modelling of systems with unpredictable or unreliable behaviour. Markov chains are a class of such probabilistic models that have been extensively used in many areas of Science and Engineering, including the analysis of performance of computer networks and of reliability of communication and security protocols [1, 2], in the study of various quantitative attributes of biochemical reaction networks [3, 4] or genetics [5]. A Markov chain can be thought of as a collection of states accompanied by a function that describes the probabilistic nature of transitions between any

[*] Corresponding author

pair of states. Depending on the type of model, these transitions can occur in discrete or continuous time. The study of such chains is carried out either analytically, or through exhaustive exploration of its execution paths, or by employing numerical schemes (usually by solving a system of equations that approximate their solution). For the analysis of continuous-time models, a typical method employed is uniformisation, which is based on a time-discretisation of the chain of interest [6]. Prominent tools allowing to analyse and verify Markov models include PRISM [7], STORM [8], or MODEST [9].

Unfortunately, efficient analysis of Markov models is difficult to achieve in practice, due to the state explosion problem. In order to enable the handling of larger state spaces, several model approximation techniques have been introduced. These techniques typically solve a smaller chain – one with a reduced state space – and then interpret results in terms of the original model. State-aggregation methods [10] construct this smaller chain by clustering (or lumping, namely aggregating) the state space. State-space truncation methods [6], on the other hand, work by dynamically neglecting states with insignificant transient probability (later we will define transient probability as the probability mass at any given time). In both cases an approximation error must be quantified. In practice, highly accurate probability approximations are crucial, for example in reliability analysis of safety-critical systems or when checking satisfiability of temporal logic formulae.

1.1. Key contributions

In this work, we present a comprehensive survey of existing state-reduction techniques for Markov chains and focus on the enhancement of a framework [10] (by the same authors) for aggregation-based analysis. In particular, the new improvements presented in this article include (1) the development of an accurate and efficient clustering (aggregation) method applicable to chains with an arbitrary structure of the underlying state space; (2) the design of more accurate bounds on the approximation error; and (3) the integration of aggregation with uniformisation (i.e. time discretisation) to enable the analysis of continuous-time models. Finally, we introduce a new hybrid scheme that utilises both aggregation and truncation of the state space and provides the best available approach for approximating continuous-time models.

In order to perform a fair experimental evaluation of the presented approximation techniques, a total of eight methods for Markov chain analysis (5 existing and 3 new ones) are implemented in the probabilistic model checker PRISM [7]. First, we showcase the developed framework for the aggregation-based analysis, highlight its improvement in both precision and speedup as compared to [10], and confirm that the resulting aggregating method can provide a valid model approximation along with precise error bounds, both in discrete and continuous time. Further, we perform the first in-depth performance evaluation of all available reduction techniques using a broad range of case studies and investigate how different methods handle various classes of models. Finally, we demonstrate that a new hybrid method significantly outperforms existing approaches and provides speedup of the analysis up to a factor of 30 with a corresponding approximation error bounded within 0.1%.

1.2. Related work

A widely studied numerical method to deal with large- or even infinite-state spaces is truncation, which works by dynamically neglecting insignificant states with a transient probability below a given threshold. This method provides under-approximation of the true probability distribution, where the probability loss serves as an error bound. In the context of continuous-time chains, a combination of truncation with adaptive uniformisation [11] is a widely used technique known as fast adaptive uniformisation (FAU) [6], where the adaptivity allows to dynamically change the uniformisation rate: this significantly decreases the number of performed uniformisations on the chain. A reduction of the state space is achieved since usually a significant percentage of the transient probability is concentrated in a small subset of states and a large fraction of the state space can be thus truncated. However, when transient probability is spread over a large number of states, these methods can result in small reduction – or poor precision if a larger probability threshold is used.

An alternative approach to achieve state-space reduction is aggregation, which typically involves clustering the state space of the model and then treating resulting clusters as abstract states of an obtained abstract chain. Analysis of this model with a reduced state space can then reveal information about the original chain. Aggregating methods include clustering based on (bi-)simulation equivalence [12], which exploits symmetries of the concrete model and performs exact numerical computation but, unfortunately, can only be applied to specific Markov processes. To enable handling of more general models, advanced notions of approximate equivalence have been introduced [13], which have led to new abstraction techniques for the numerical analysis of Markov chains with finite [14] as well as with uncountably infinite state spaces [15, 16]. The work in [17] presents an algorithm to approximate probability distributions of a Markov model forward in time, which inspired the adaptive scheme proposed in [10], where a formal error analysis steers the adaptation. This use of derived error bounds allows far greater accuracy and flexibility, as it accounts also for the history of the transient probability within specific clusters.

Biochemical systems, in particular, have given rise to a number of specialised methods for analysis. Such systems are described using the language of Chemical Reaction Networks (CRNs), the time-evolution of which is governed by the Chemical Master Equation (CME) [18]. To enable the analysis of networks with large populations of species or those exhibiting complex dynamics, various approximation techniques have been proposed. These include, for example, the Linear Noise Approximation [14], where a Gaussian process approximates a CME and describes the time evolution of expectation and variance of the species in terms of a set of ordinary differential equations (ODEs). Recently, an aggregation scheme over ODEs that aims at understanding the dynamics of large CRNs has been proposed in [19]. These deterministic approximations, however, cannot adequately capture the stochasticity of CRNs caused by low population species. In order to cope with such systems, various hybrid models have been proposed [20, 21], where the dynamics of low population species are encompassed by a discrete stochastic process and the dynamics of large population species is approximated by a continuous one. These methods, however, do not provide any formal guarantees on the approximation error. In [22], a novel semi-quantitative framework for the analysis of CRNs has been introduced which offers greater performance and scalability compared to existing techniques, although a formal guarantee is missing.

Simulation-based methods qualitatively analyse the behaviour of a Markov chain by generating its trajectories: collecting the statistics from multiple realisations then allows to estimate the transient probability distribution. Simulation-based techniques include e.g. Gillespie's Stochastic Simulation Algorithm [23] as well as its various modifications [24, 25, 26, 27]. These methods do not provide formal bounds on the error and instead can give weak precision guarantees in the form of confidence intervals. Furthermore, a large number of simulations is required to provide precise results, which can be very time consuming. Nonetheless, these practical approaches are suitable for situations where highly accurate estimates of the transient probability distribution are not necessary.

1.3. Structure of the article

In Section 2 we present an overview of existing approximate techniques for the analysis of Markov chains, namely, state-space aggregation and truncation, and review the uniformisation technique, including its fast adaptive version. In Section 3 we redefine a notion of a state-space aggregation of a discrete-time Markov chain, introduce algorithms that allow us to approximate chains with arbitrary structure of the state space, explore various aggregation strategies and design a more precise approximation error bound. Then, in Section 4, we combine these results with uniformisation techniques to enable handling continuous-time models and introduce a new hybrid scheme that utilises both aggregation and truncation of the state space. Finally, in Section 5 we discuss experimental results.

2. Preliminaries

In this section, we review the necessary theory and introduce notation that will be used throughout the paper. First, we discuss discrete-time Markov chains, describe state-space truncation and summarise an aggregation approach presented in [10]. Then we consider continuous-time models and describe the uniformisation technique used for their analysis.

2.1. Discrete-time Markov chains

Definition 1. *A discrete-time Markov chain (DTMC)* is a pair $D = (S, P)$, where

- S is *the state space*, and

- $P\colon S \times S \to [0,1], \forall r \in S\colon \sum_{s \in S} P(r,s) = 1$, is *the transition probability matrix.*

Unless stated otherwise, we assume that the state space S is finite. The model is initialised via the probability distribution $p_0\colon S \to [0,1]$, $\sum_{s \in S} p_0(s) = 1$, and its transient probability distribution at time step $k > 0$ is

$$p_k(s) = \sum_{r \in S} p_{k-1}(r) P(r,s), \tag{1}$$

or, using matrix notation, $p_k = p_{k-1} \cdot P$. The act of performing one such multiplication is called *an iteration*, *a probability propagation*, or *a discrete-time step.* The problem of finding transient probability distribution p_k at an arbitrary time horizon $k \geq 0$ is referred to as *the transient analysis* of the chain. Computing this vector directly using (1) suffers from the state explosion problem, and we are therefore interested in providing an efficient and accurate approximation.

2.2. State-space aggregation of DTMC

In this subsection, we briefly describe the state-space aggregation method, as it was presented in [10]. We omit some technical details and discuss them later in Section 3, where the generalised framework for state-space aggregation is developed.

Let $D = (S,P)$ be a DTMC initialised via the probability distribution p_0, and assume that we are interested in approximating its transient probability distribution $p_k,\ k > 0$. Let Φ be a partition of the state space S. We treat clusters in Φ as abstract states of a new aggregated chain $\Delta = (\Phi, \Pi)$, where Π represents a suitable abstract transition probability matrix $\Pi\colon \Phi \times \Phi \to \mathbb{R}_{\geq 0}$. The DTMC Δ is initialised using the probability distribution $\pi_0\colon \Phi \to \mathbb{R}_{\geq 0}$, computed as

$$\pi_0(\sigma) = \sum_{s \in \sigma} p_0(s), \tag{2}$$

i.e. the probability of residing in cluster σ is the sum of transient probabilities for the concrete states in σ. We can now recursively compute the transient probability distribution π_k of Δ using vector-matrix multiplications, similar to (1), as:

$$\pi_k = \pi_{k-1} \cdot \Pi. \tag{3}$$

Having obtained π_k, i.e. the probability of residing in individual clusters at time k, the approximation $\tilde{p}_k\colon S \to \mathbb{R}_{\geq 0}$ of the transient probability distribution p_k is defined as

$$\tilde{p}_k(s) = \frac{\pi_k(\sigma)}{|\sigma|}, s \in \sigma, \tag{4}$$

where $|\sigma|$ denotes the size of cluster σ (namely the number of concrete states comprising it). That is, the probability of residing in cluster σ is distributed uniformly among its concrete states.

If we select a partition Φ such that $|\Phi| \ll |S|$, working with the abstract chain (Φ, Π) and performing vector-matrix multiplications (3) allows to approximate p_k using π_k, and to reduce the computational demands when compared to performing the discrete steps in (1) with the concrete model. Let $e_k \coloneqq \tilde{p}_k - p_k$ denote an error vector associated with this approximation. We cannot compute this vector directly since we do not know the exact probabilities p_k. However, we can formally bound the L_1-norm of this vector: this norm is often used with stochastic vectors and is efficiently computable from the structure of (Φ, Π). Introduce the quantity

$$\epsilon(\rho, \sigma) \coloneqq \max_{s \in \sigma} \left| \Pi(\rho, \sigma) - \frac{|\sigma|}{|\rho|} \sum_{r \in \rho} P(r, s) \right| \tag{5}$$

and further denote $\epsilon(\rho) \coloneqq \sum_{\sigma \in \Phi} \epsilon(\rho, \sigma)$. Then for the L_1-norm of the error vector at time $k > 0$ it holds that

$$\|e_k\|_1 \leq \|e_{k-1}\|_1 + \sum_{\rho \in \Phi} \pi_{k-1}(\rho) \cdot \epsilon(\rho), \tag{6}$$

where

$$\|e_0\|_1 = \sum_{s \in S} |p_0(s) - \tilde{p}_0(s)|. \tag{7}$$

The term $\|e_0\|_1$ is called *aggregation error* and describes the inaccuracy introduced when the exact p_0 is replaced with $\tilde{p}_0$. Additionally, during each discrete step, a *propagation error*, associated with the use of the abstraction Π instead of P, is produced and is captured by $\epsilon(\cdot, \cdot)$: this quantity accounts for the maximum difference, for a given pair of clusters, between the abstract transition probability and (rescaled) incoming probability. The product $\pi_{k-1}(\rho) \cdot \epsilon(\rho)$ provides an upper bound on the error generated from ρ; the sum over all abstract states in (6) then yields the overall error. The reason for computing the bound on the L_1-norm of e_k is that, as was already mentioned, we want to reduce the computational complexity of the error estimation: Equation (6) suggests that one step of this estimation is equivalent to performing a scalar product of two vectors in the aggregated (i.e. with the reduced state space) setting.

So far, we have ignored how the partition Φ of S and how the abstract transition probability matrix Π are computed. In [10], where this approach is used to analyse biochemical processes, the aggregation is constructed based on the known structure of the model and on the underlying physics. Later, in Section 3, we will develop a generalized approach and describe how to aggregate chains with an arbitrary structure of their state space – until then, assume that a specific clustering is given. Eventually we will see that, as we perform discrete steps in the abstract setting, the probability distribution shifts, so adapting the clustering of the state space can reduce the error: *an adaptive state-space aggregation* is therefore a method of using different clusterings sequentially in time, where the quality of each clustering is quantified by (6).

Remark. On a final note, observe that we do not require the normalisation condition $\forall \rho \in \Phi\colon \sum_{\sigma \in \Phi} \Pi(\rho, \sigma) = 1$ to hold: in other words, Π might not be a stochastic matrix, see Figure 1. In this case, elements of vectors π_k might not sum to one and therefore such vectors cannot be called 'probability vectors'. Thus, given Π, abstraction (Φ, Π) might not be viewed as an actual DTMC according to the Definition 1. However, we still want to associate elements of Π with transition probabilities and elements of π_k with transient probabilities of the aggregated model. To avoid any confusion, we will reserve the term 'stochastic' for matrices and vectors that satisfy the normalisation property. Lack of stochasticity should not discourage us from using such abstractions: function Π (respectively, π_k) simply serves as a higher-level representative of function P (respectively, p_k) on a new state space and provides us with an approximation of its concrete counterpart. □

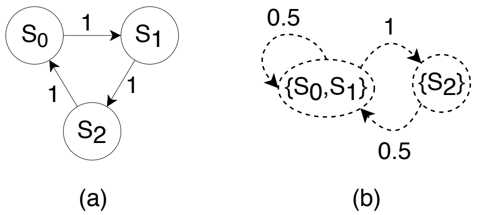

Figure 1: In (a) we consider a simple DTMC with three states; in (b) we construct its aggregation based on the partition $\Phi = \{\{s_0, s_1\}, \{s_2\}\}$, where the abstract transition matrix Π is computed using (18) – notice that in this case Π is not stochastic.

2.3. State-space truncation of DTMC

State-space truncation [6] is yet another approximation technique that can be applied to any Markov chain. Its transient probability distribution is computed similarly to (1), however before each iteration, states with insignificant transient probability, i.e. those with probability below some specified truncation threshold δ_{tru}, are discarded. The effect of this truncation is twofold. First of all, we are now able to propagate probability mass only from *significant* (also called *active*) states, which leads to a speedup of the analysis. On the other hand, each truncated state with a non-zero transient probability contributes to the probability loss and the resulting transient probability distribution $\tilde{p}_k$ is actually an under-approximation of the true distribution p_k: the probability mass that was truncated could have remained in a given state or might have been transported to other ones. The total probability loss $1 - \sum_{s \in S} \tilde{p}_k(s)$ serves as an exact upper bound on the L_1-norm of the error vector. The truncation threshold

δ_{tru} specifies a trade-off between performance and accuracy: large values of δ_{tru} will discard a considerable number of states for the price of reduced precision, and vice versa. Note that this approach can also be useful for chains with an infinite state space, because at each iteration we can deal only with the (finite) set of active states. This technique fails, however, for chains with slow dynamics of the transient probability: inactive states that slowly receive small accruals of the probability mass never become active if the truncation threshold is not small enough, which can yield an enormous error. In general, this does not happen with the state-space aggregation: working with clusters preserves – at least to some extent – information about where the residual transient probability is located. On the other hand, ϵ-terms might result in rather conservative error bounds, when compared to the actual probability lost from the transitions.

2.4. Continuous-time Markov chains

Definition 2. *A continuous-time Markov chain (CTMC)* is a pair $C = (S, R)$, where

- S is *the set of states*, and
- $R\colon S \times S \to \mathbb{R}_{\geq 0}, \forall s \in S\colon R(s, s) = 0$, is *the transition rate matrix.*

Similar to DTMCs, we will implicitly assume that the set S is finite. R is a function that for each pair of different states assigns a rate used as a parameter of an exponential distribution when deriving a probability of transitioning between these states within a continuous time window. The time spent in state r, before any such transition occurs, is exponentially distributed with parameter $E(r) := \sum_{s\in S} R(r, s)$, which is called *the exit rate* of state r. An *infinitesimal generator matrix* associated with C is a matrix $Q\colon S \times S \to \mathbb{R}$ defined as $Q(s, s) = -E(s)$ and $Q(r, s) = R(r, s)$ for $r \neq s$.

Definition 3. Let $C = (S, R)$ be a CTMC and Q be its infinitesimal generator matrix. Let $q \geq \max_{s\in S} E(s)$ be *a uniformisation rate. A uniformised DTMC of C given uniformisation rate q* is a DTMC (S, unif^q_R) having its transition probability matrix unif^q_R defined as

$$\mathrm{unif}^q_R(r, s) = \begin{cases} 1 + \frac{Q(r,s)}{q}, & \text{if } r = s \\ \frac{Q(r,s)}{q}, & \text{otherwise.} \end{cases}$$

A uniformised DTMC serves as a time-discretisation of a CTMC with respect to the fastest event that can occur with rate q. Usually, we select q to be the maximum exit rate from states in S, which corresponds to the shortest mean residence time of the model, although any value larger or equal to this rate is also usable.

A CTMC is initialised via the probability distribution $p_0\colon S \to [0, 1]$. The transient probability distribution p_t at an arbitrary time horizon $t \in \mathbb{R}_{\geq 0}$ can be obtained by employing a uniformisation technique. The main idea of this method is to split a CTMC $C = (S, R)$ into two independent stochastic processes

D_C and B_C: D_C is a DTMC over the same state space S; B_C is a CTMC (called *a birth process of* C) having an infinite state space $\mathbb{N}_0$, such that

$$p_t(s) = \sum_{k=0}^{\infty} u_k(s) \cdot \beta_t(k), \tag{8}$$

where u_k denotes the transient probability distribution of D_C at time k and $\beta_t(\cdot)$ denotes the transient probability distribution of B_C at time t. Intuitively, Equation (8) invokes the total probability theorem on the number of discrete 'jumps' of C taking place up to time t: the quantity $u_k(s)$ represents the probability of residing in state s after exactly k discrete jumps, whereas $\beta_t(k)$ expresses the probability of actually performing these k jumps within a continuous window of length t. Under this interpretation, D_C keeps track of the current state of C and B_C keeps track – in a probabilistic sense – of whether time t has elapsed.

2.5. Standard uniformisation of CTMC

The choice of D_C and B_C is key for the uniformisation procedure. A *standard uniformisation (SU)* works by selecting a uniformisation rate $q \geq \max_{s \in S} E(s)$, and then choosing D_C to be a uniformised DTMC (S, unif_R^q) with rate q, and B_C to be a pure birth process with constant rate q (i.e. Poisson with rate q), for which the analytical solution is known to be $\beta_t(k) = e^{-qt} \frac{(qt)^k}{k!} =: \psi_{qt}(k)$, $k \in \mathbb{N}_0$. Finally, for a given precision (i.e., maximum error) ε_{fg}, the iterative scheme of Fox and Glynn [28] can provide parameters L, R such that

$$1 - \varepsilon_{fg} \leq \sum_{k=L}^{R} \psi_{qt}(k). \tag{9}$$

We can then truncate the infinite sum in (8) to obtain an under-approximation $\hat{p}_t$ of the true probability distribution p_t, as:

$$\hat{p}_t(s) = \sum_{k=L}^{R} u_k(s) \cdot \psi_{qt}(k) \leq p_t(s). \tag{10}$$

Combining (9) and (10), we arrive at

$$\begin{aligned} \|\hat{p}_t\|_1 &= \sum_{s \in S} \hat{p}_t(s) = \sum_{s \in S} \sum_{k=L}^{R} u_k(s) \cdot \psi_{qt}(k) = \sum_{k=L}^{R} \sum_{s \in S} u_k(s) \cdot \psi_{qt}(k) \\ &= \sum_{k=L}^{R} \psi_{qt}(k) \cdot \sum_{s \in S} u_k(s) = \sum_{k=L}^{R} \psi_{qt}(k) \geq 1 - \varepsilon_{fg}, \end{aligned}$$

and therefore

$$\|e_t\|_1 = \|p_t - \hat{p}_t\|_1 = \|p_t\|_1 - \|\hat{p}_t\|_1 = 1 - \|\hat{p}_t\|_1 \leq \varepsilon_{fg},$$

as desired. In other words, we have a guarantee that the total probability loss resulting from the truncation of the infinite sum in (8) will not exceed ε_{fg}. Note that this proposition holds since the transient probabilities $u_k(s)$ sum to one. The main drawback of SU is that for large uniformisation rates q the mean of the Poisson distribution $\psi_{qt}(\cdot)$ is large and so is the upper truncation point R. This implies that, in order to obtain $\hat{p}_t$, one must perform plenty of iterations for the uniformised DTMC D_C.

2.6. Adaptive uniformisation of CTMC

As an alternative to SU, *adaptive uniformisation (AU)* [11] allows the rates of the birth process to change at each step, according to the following rules:

- let $q_0, q_1, ...$ be an infinite sequence of uniformisation rates satisfying

$$q_i \geq \max\{E(s) \mid s \in S, u_i(s) > 0\}; \tag{11}$$

- let $B_C = (\mathbb{N}_0, R_{B_C})$ be a CTMC starting at state 0 and R_{B_C} be defined as

$$R_{B_C}(i,j) = \begin{cases} q_i, & \text{if } j = i+1 \\ 0, & \text{otherwise;} \end{cases}$$

- let $D_C = (S, \mathrm{unif}_R^{q_i})$ be a DTMC with transition probability matrix at step i to be a uniformisation of R with rate q_i, and where u_i denotes its transient probability distribution at time step i.

The analysis of a CTMC C via adaptive uniformisation proceeds as follows. We start at a discrete time 0 with a subset of states in S that have non-zero initial transient probability $u_0(\cdot)$ – such states are called *significant* or *active*. The largest exit rate among the states within this subset is the (local) uniformisation rate q_0. We then compute $\mathrm{unif}_R^{q_0}$ to be the transition probability matrix for the process D_C at (discrete) time 0, perform probability propagation and obtain $u_1(\cdot)$. We then repeat the procedure of defining the subset of active states, finding a (local) uniformisation rate, uniformizing the rate matrix according to this rate and propagating the probability. This way we obtain a sequence $u_0, u_1, \ldots$, that will be used to compute p_t according to (8), where $\beta_t(\cdot)$ is a solution to the birth process B_C with rates $q_0, q_1, \ldots$. In order to solve this CTMC, we apply SU using a (global) uniformisation rate $q \geq max_{s \in S} E(s)$, since $\forall i \in \mathbb{N}_0$, $q_i \leq q$. Together with the Fox-Glynn method for a given precision ε_{fg}, we can compute an under-approximation $\hat{\beta}_t$ of the transient probabilities β_t at time t of the birth process B_C. The resulting under-approximation of the transient probability distribution $\hat{p}_t$ is then computed as

$$\hat{p}_t(s) = \sum_{k=0}^{R'} u_k(s) \cdot \hat{\beta}_t(k),$$

where R' is a time step for which $\sum_{k=0}^{R'} \hat{\beta}_t(k) \geq 1 - \varepsilon_{bp}$ for a given precision $\varepsilon_{bp} < \varepsilon_{fg}$. Both truncations – for uniformising C and uniformising the inner birth process B_C – lead to an under-approximation $\hat{p}_t$ of the true probability distribution p_t, and the total error is given by the probability loss $1 - \|\hat{p}_t\|_1$ with a bound ε_{bp} specified a-priori.

The main advantage of AU is that uniformisation rates q_i are 'discovered' as the probability propagates. As there is a chance that at any given time i, q_i will be substantially lower than q, this allows the birth process to jump at lower rates, and therefore that R' is substantially lower than the corresponding upper Fox-Glynn limit R in the case of SU. Also note that, due to its extremely simple structure, solving for the birth process during each iteration is a trivial task. Numerically, this allows to perform much fewer vector-matrix multiplications to solve for D_C and to obtain the same result with the same precision as SU.

2.7. Beyond uniformisations of CTMC: combined approximations

State-space truncation can be used with both SU or AU to solve for D_C, although it is particularly favorable with AU since truncating the state space using threshold $\delta_{tru} > 0$ leads to smaller subsets of active states and the birth process B_C can jump at even slower rates q_i. In the sequel, we will refer to the combination of SU with state-space truncation as *fast SU (FSU)*, and to the combination of AU with truncation as *fast AU (FAU)* [6]. In both of these cases, however, the probability loss is the only way to compute the approximation error, since an a-priori specified error bound cannot be guaranteed.

Finally, SU can be integrated with the state-space aggregation approach while solving for uniformised DTMC. This idea was first outlined in [10], although a rigorous formalisation is still missing and will be the focus of Section 4. There we shall also explore combinations of aggregation and truncation approaches ($FAU+$ hybrid scheme).

3. State-Space Aggregation of DTMCs

In this section, we develop ideas presented in [10] and formalise a notion of a general state-space aggregation for discrete-time Markov chains. This will allow to explore various aggregation schemes as well as to obtain more precise bounds on the approximation errors. Finally, we describe an aggregation procedure that is capable of handling arbitrary DTMCs. In Section 4 we will then apply these ideas to enable the analysis of continuous-time models.

3.1. Approximate DTMCs

We wish to explore how altering the transition probability matrix of a DTMC affects its transient behaviour. We will do so by introducing the concept of an approximate DTMC.

Definition 4. Let $D = (S, P)$ be a DTMC. Let $\tilde{P} \colon S \times S \to \mathbb{R}$ be an approximation of P with the same dimensions. A pair $(S, \tilde{P})$ is referred to as *an approximation of the DTMC D.*

Assume p_0 and $\tilde{p}_0$ are initial transient probability distributions of (S, P) and $(S, \tilde{P})$, respectively. Here we interpet $\tilde{p}_0$ as an approximation of p_0. We can now compute the transient probability distribution $\tilde{p}_k$ of the approximate DTMC $(S, \tilde{P})$ at any time $k > 0$ similarly to propagation Equation (1):

$$\tilde{p}_k = \tilde{p}_{k-1} \cdot \tilde{P}. \tag{12}$$

The transient probability distribution $\tilde{p}_k,\ k \geq 0$, serves as an approximation of the distribution p_k. We are interested in computing an approximation error $e_k \coloneqq \tilde{p}_k - p_k$. First, observe that e_0 captures the error associated with approximating p_0 by $\tilde{p}_0$. For $k > 0$, it holds that

$$\begin{aligned} p_k + e_k = \tilde{p}_k = \tilde{p}_{k-1} \cdot \tilde{P} &= \tilde{p}_{k-1} \cdot \left(\tilde{P} - P + P\right) = \tilde{p}_{k-1} \cdot P + \tilde{p}_{k-1} \cdot \left(\tilde{P} - P\right) \\ &= (p_{k-1} + e_{k-1}) \cdot P + \tilde{p}_{k-1} \cdot \left(\tilde{P} - P\right) \\ &= p_{k-1} \cdot P + e_{k-1} \cdot P + \tilde{p}_{k-1} \cdot \left(\tilde{P} - P\right) \\ &= p_k + e_{k-1} \cdot P + \tilde{p}_{k-1} \cdot \left(\tilde{P} - P\right), \end{aligned}$$

which implies

$$e_k = e_{k-1} \cdot P + \tilde{p}_{k-1} \cdot \left(\tilde{P} - P\right).$$

This recursion provides insight into how an error is generated when we approximate a DTMC. The first summand represents propagation of an existing error as if we were using exact transitions P for probability propagation. On the other hand, the term $\tilde{p}_{k-1} \cdot \left(\tilde{P} - P\right)$ captures an error that is generated at each step between each pair of states while using the approximation $\tilde{p}_{k-1}$ for p_{k-1} and $\tilde{P}$ for P. Recall that we are interested in the L_1-norm of the error vector e_k, namely:

$$\begin{aligned} \left\|e_k\right\|_1 &= \left\|e_{k-1} \cdot P + \tilde{p}_{k-1} \cdot \left(\tilde{P} - P\right)\right\|_1 \\ &\leq \left\|e_{k-1} \cdot P\right\|_1 + \left\|\tilde{p}_{k-1} \cdot \left(\tilde{P} - P\right)\right\|_1. \end{aligned}$$

One can easily argue that $\left\|v \cdot A\right\|_1 \leq \left\|v\right\|_1$ for any vector v and any stochastic matrix A, and therefore expression above becomes

$$\left\|e_k\right\|_1 \leq \left\|e_{k-1}\right\|_1 + \left\|\tilde{p}_{k-1} \cdot \left(\tilde{P} - P\right)\right\|_1. \tag{13}$$

3.2. State-space aggregation of DTMCs

Let us now introduce a specific class of approximate DTMCs: we will reinterpret the above derivation accordingly.

Definition 5. Let $D = (S, P)$ be a DTMC with initial probability distribution p_0 and let Φ be a clustering of S. Recall that for a cluster $\sigma \in \Phi$, $|\sigma|$ denotes the cardinality of set σ, i.e. number of concrete states that comprise this cluster. Let $\Pi\colon \Phi \times \Phi \to \mathbb{R}$ be any real-valued matrix relating pairs of clusters and let $\pi_0\colon \Phi \to \mathbb{R}$ be a row vector s.t. $\pi_0(\sigma) \coloneqq \sum_{s\in\sigma} p_0(s)$. An approximation $(S, \tilde{P})$ of the DTMC D, where $\tilde{P}(r, s) = \Pi(\rho, \sigma)/|\sigma|, r \in \rho, s \in \sigma$, with initial distribution $\tilde{p}_0(s) = \pi_0(\sigma)/|\sigma|, s \in \sigma$, will be referred to as *a state-space aggregation of D* given *an abstract state space* Φ and *an abstract transition probability function* Π.

Notice that $\forall s, s' \in \sigma$, it holds that $\tilde{p}_0(s) = \frac{\pi_0(\sigma)}{|\sigma|} = \tilde{p}_0(s')$. Similarly, for $(r, r' \in \rho)$ and $(s, s' \in \sigma)$ we have that $\tilde{P}(r, s) = \frac{\Pi(\rho,\sigma)}{|\sigma|} = \tilde{P}(r', s')$, and therefore the update Equation (12) yields, for all $s, s' \in \sigma$,

$$\tilde{p}_k(s) = \sum_{r\in S} \tilde{p}_{k-1}(r) \cdot \tilde{P}(r, s) = \sum_{r\in S} \tilde{p}_{k-1}(r) \cdot \tilde{P}(r, s') = \tilde{p}_k(s').$$

The three equalities above illustrate an important property of the aggregation: any two states from two given clusters have the same approximate transition probability $\tilde{P}(\cdot, \cdot)$ and any two states in a given cluster at any time step $k \geq 0$ share the same approximate transient probability $\tilde{p}_k(\cdot)$. Introduce the quantity $\pi_k(\sigma) \coloneqq \sum_{s\in\sigma} \tilde{p}_k(s)$, $\sigma \in \Phi$, $k > 0$. From the argument above it follows that $\tilde{p}_k(s) = \pi_k(\sigma)/|\sigma|$ for any $s \in \sigma$. Intuitively, the quantity $\pi_k(\sigma)$ describes probability of residing in one of the states in cluster σ at time step k, whereas $\Pi(\rho, \sigma)$ captures the probability of transitioning into one of the states in cluster σ given that the chain resides in one of the states in cluster ρ.

Finally, for $k > 0$, it holds that:

$$\begin{aligned}
\pi_k(\sigma) &= \sum_{s\in\sigma} \tilde{p}_k(s) = \sum_{s\in\sigma}\sum_{r\in S} \tilde{p}_{k-1}(r) \cdot \tilde{P}(r, s) = \sum_{s\in\sigma}\sum_{\rho\in\Phi}\sum_{r\in\rho} \tilde{p}_{k-1}(r) \cdot \tilde{P}(r, s) \\
&= \sum_{s\in\sigma}\sum_{\rho\in\Phi}\sum_{r\in\rho} \frac{\pi_{k-1}(\rho)}{|\rho|} \cdot \frac{\Pi(\rho, \sigma)}{|\sigma|} = \sum_{\rho\in\Phi}\sum_{s\in\sigma}\sum_{r\in\rho} \frac{\pi_{k-1}(\rho)}{|\rho|} \cdot \frac{\Pi(\rho, \sigma)}{|\sigma|} \\
&= \sum_{\rho\in\Phi} \pi_{k-1}(\rho) \cdot \Pi(\rho, \sigma) \cdot \frac{1}{|\rho| \cdot |\sigma|} \sum_{s\in\sigma}\sum_{r\in\rho} 1 = \sum_{\rho\in\Phi} \pi_{k-1}(\rho) \cdot \Pi(\rho, \sigma),
\end{aligned}$$

or in matrix notation:

$$\pi_k = \pi_{k-1} \cdot \Pi. \tag{14}$$

Hence, instead of computing approximate transient probabilities for each of the states, we can first aggregate the initial probability distribution p_0 into π_0 according to the definition of the latter, compute π_k using (14) and then find $\tilde{p}_k(s)$ as $\pi_k(\sigma)/|\sigma|$ for any state s. A pair (Φ, Π) might be interpreted as a DTMC (where the transition matrix Π is not necessarily stochastic) representing the aggregation of (S, P) and providing the approximation of its probability distribution. As to an error associated with this approximation, we use properties of aggregation to rewrite (13) into (16): the second term in (13) reads:

$$\left\|\tilde{p}_{k-1} \cdot \left(\tilde{P} - P\right)\right\|_1 = \sum_{s \in S} \left| \sum_{r \in S} \tilde{p}_{k-1}(r) \cdot \left(\tilde{P}(r,s) - P(r,s)\right) \right| .$$

Running both summations over clusters and then applying the observation above concerning states within the same cluster yields

$$\begin{aligned}
&\sum_{\sigma \in \Phi} \sum_{s \in \sigma} \left| \sum_{\rho \in \Phi} \sum_{r \in \rho} \tilde{p}_{k-1}(r) \cdot \left(\tilde{P}(r,s) - P(r,s)\right) \right| \\
&\quad = \sum_{\sigma \in \Phi} \sum_{s \in \sigma} \left| \sum_{\rho \in \Phi} \sum_{r \in \rho} \frac{\pi_{k-1}(\rho)}{|\rho|} \cdot \left(\frac{\Pi(\rho,\sigma)}{|\sigma|} - P(r,s) \right) \right| \\
&\quad = \sum_{\sigma \in \Phi} \sum_{s \in \sigma} \left| \sum_{\rho \in \Phi} \pi_{k-1}(\rho) \sum_{r \in \rho} \left(\frac{\Pi(\rho,\sigma)}{|\rho| \cdot |\sigma|} - \frac{1}{|\rho|} P(r,s) \right) \right| \\
&\quad = \sum_{\sigma \in \Phi} \sum_{s \in \sigma} \left| \sum_{\rho \in \Phi} \pi_{k-1}(\rho) \left(\frac{\Pi(\rho,\sigma)}{|\sigma|} - \frac{1}{|\rho|} \sum_{r \in \rho} P(r,s) \right) \right| .
\end{aligned}$$

The following are algebraic manipulations with summations and absolute values:

$$\begin{aligned}
&\sum_{\sigma \in \Phi} \sum_{s \in \sigma} \left| \sum_{\rho \in \Phi} \pi_{k-1}(\rho) \left(\frac{\Pi(\rho,\sigma)}{|\sigma|} - \frac{1}{|\rho|} \sum_{r \in \rho} P(r,s) \right) \right| \\
&\quad \leq \sum_{\sigma \in \Phi} \sum_{s \in \sigma} \sum_{\rho \in \Phi} \pi_{k-1}(\rho) \left| \frac{\Pi(\rho,\sigma)}{|\sigma|} - \frac{1}{|\rho|} \sum_{r \in \rho} P(r,s) \right| \\
&\quad = \sum_{\rho \in \Phi} \sum_{\sigma \in \Phi} \sum_{s \in \sigma} \pi_{k-1}(\rho) \left| \frac{\Pi(\rho,\sigma)}{|\sigma|} - \frac{1}{|\rho|} \sum_{r \in \rho} P(r,s) \right| \\
&\quad = \sum_{\rho \in \Phi} \pi_{k-1}(\rho) \sum_{\sigma \in \Phi} \sum_{s \in \sigma} \left| \frac{\Pi(\rho,\sigma)}{|\sigma|} - \frac{1}{|\rho|} \sum_{r \in \rho} P(r,s) \right| .
\end{aligned}$$

Based on the obtained formulation, let us now define

$$\tau(\rho,\sigma) \coloneqq \sum_{s\in\sigma} \left| \frac{\Pi(\rho,\sigma)}{|\sigma|} - \frac{1}{|\rho|}\sum_{r\in\rho} P(r,s) \right|, \tag{15}$$

and $\tau(\rho) \coloneqq \sum_{\sigma\in\Phi} \tau(\rho,\sigma)$. Then for the L_1-norm of the error vector e_k it holds that

$$\|e_k\|_1 \leq \|e_{k-1}\|_1 + \sum_{\rho\in\Phi} \pi_{k-1}(\rho)\cdot\tau(\rho). \tag{16}$$

Note the similarity of the equation above with (6). Furthermore, observe that

$$\begin{aligned}\tau(\rho,\sigma) &= \sum_{s\in\sigma} \left| \frac{\Pi(\rho,\sigma)}{|\sigma|} - \frac{1}{|\rho|}\sum_{r\in\rho} P(r,s) \right| \leq |\sigma|\cdot \max_{s\in\sigma} \left| \frac{\Pi(\rho,\sigma)}{|\sigma|} - \frac{1}{|\rho|}\sum_{r\in\rho} P(r,s) \right| \\ &= \max_{s\in\sigma} \left| \Pi(\rho,\sigma) - \frac{|\sigma|}{|\rho|}\sum_{r\in\rho} P(r,s) \right| = \epsilon(\rho,\sigma),\end{aligned}$$

and therefore the τ-terms in (16) provide a better error bound than that in (6) based on ϵ-terms.

3.3. Specific forms of state-space aggregation

So far we have assumed nothing about the form of Π: error bound (16) allows arbitrary abstract transition probability matrices. Recall that the matrix Π determines the structure of the approximate transition matrix $\tilde{P}$. Therefore, we wish to construct Π based on the knowledge of P so that the resulting approximation $\tilde{P}$ is as precise as possible. Let us now be more specific and describe three possible ways to define Π.

3.3.1. State-space aggregation based on average incoming transition probabilities

The approximate transition matrix for this aggregation is defined as follows:

$$\Pi_{in}(\rho,\sigma) = \frac{1}{|\sigma|}\sum_{r\in\rho}\sum_{s\in\sigma} P(r,s). \tag{17}$$

The intuition behind this equation is that it encompasses the average *incoming* probability to cluster σ from cluster ρ. This shape of the transition matrix Π was previously introduced in [10] and the error bound (6) based on ϵ-terms was derived accordingly.

3.3.2. State-space aggregation based on average outgoing transition probabilities

This scheme is similar to the previous one, except that now we utilise average outgoing transition probabilities:

$$\Pi_{out}(\rho, \sigma) = \frac{1}{|\rho|} \sum_{r \in \rho} \sum_{s \in \sigma} P(r, s). \tag{18}$$

Here we demonstrate that outgoing averaging (18) is a natural approach to define transitions between clusters of states. Let $S_k \in S$ be a random variable describing the current state of the DTMC at time $k \geq 0$. Likewise, let $C_k \in \Phi$ be a random variable describing in which of the clusters the DTMC resides at time k. Note that, by definition, $P(r, s) = \mathbb{P}(S_{k+1} = s \mid S_k = r)$ and that $\Pi(\rho, \sigma) = \mathbb{P}(C_{k+1} = \sigma \mid C_k = \rho)$. Also, recall that, given that the chain resides in cluster σ, the probability of residing in any of the state $s \in \sigma$ is uniformly distributed between the states. We obtain

$$\begin{aligned}
\Pi(\rho, \sigma) &= \mathbb{P}(C_{k+1} = \sigma \mid C_k = \rho) = \sum_{r \in \rho} \mathbb{P}(C_{k+1} = \sigma, S_k = r \mid C_k = \rho) \\
&= \sum_{r \in \rho} \mathbb{P}(C_{k+1} = \sigma \mid C_k = \rho, S_k = r) \cdot \mathbb{P}(S_k = r \mid C_k = \rho) \\
&= \sum_{r \in \rho} \mathbb{P}(C_{k+1} = \sigma \mid S_k = r) \cdot \frac{1}{|\rho|} = \frac{1}{|\rho|} \cdot \sum_{r \in \rho} \mathbb{P}(C_{k+1} = \sigma \mid S_k = r) \\
&= \frac{1}{|\rho|} \cdot \sum_{r \in \rho} \sum_{s \in \sigma} \mathbb{P}(C_{k+1} = \sigma, S_{k+1} = s \mid S_k = r) \\
&= \frac{1}{|\rho|} \cdot \sum_{r \in \rho} \sum_{s \in \sigma} \mathbb{P}(S_{k+1} = s \mid S_k = r) = \frac{1}{|\rho|} \cdot \sum_{r \in \rho} \sum_{s \in \sigma} P(r, s),
\end{aligned}$$

which is $\Pi_{out}(\rho, \sigma)$. As a consequence, the transition probability function Π_{out} has another property, namely that, for each cluster ρ,

$$\begin{aligned}
\sum_{\sigma \in \Phi} \Pi_{out}(\rho, \sigma) &= \sum_{\sigma \in \Phi} \frac{1}{|\rho|} \sum_{r \in \rho} \sum_{s \in \sigma} P(r, s) = \frac{1}{|\rho|} \sum_{r \in \rho} \sum_{\sigma \in \Phi} \sum_{s \in \sigma} P(r, s) \\
&= \frac{1}{|\rho|} \sum_{r \in \rho} \sum_{s \in S} P(r, s) = \frac{1}{|\rho|} \sum_{r \in \rho} 1 = 1,
\end{aligned}$$

i.e. matrix Π_{out} is *stochastic*, unlike general abstract matrices Π. Since p_0 is always a stochastic vector, then so is π_0, by definition. This implies that all vectors π_k and therefore all $\tilde{p}_k$ are stochastic as well. So, the abstract chain (Φ, Π_{out}) is actually a DTMC, as per Definition 1. This subtle difference has two benefits. First, from the technical standpoint, this leads to a slightly better approximation compared to the incoming-based scheme (17), as will be shown later. Second, preserving stochasticity of $\tilde{p}_k$ will be the key to integrating aggregation technique with the uniformisation method (see Section 4), in order to enable the aggregating analysis of CTMCs.

3.3.3. Median-based state-space aggregation

This scheme is defined as follows:

$$\Pi_{med}(\rho, \sigma) = \frac{|\sigma|}{|\rho|} \operatorname*{med}_{s \in \sigma} \left\{ \sum_{r \in \rho} P(r, s) \right\}. \tag{19}$$

The median-based scheme was derived using the following argument. Assume a specific state-space clustering is given. We can arbitrarily define the abstract transition probabilities $\Pi(\cdot, \cdot)$ and the approximation error accrual in each iteration is captured by (16). In order to minimize this accrual, it is sufficient to minimize each of the terms

$$\tau(\rho, \sigma) = \sum_{s \in \sigma} \left| \frac{\Pi(\rho, \sigma)}{|\sigma|} - \frac{1}{|\rho|} \sum_{r \in \rho} P(r, s) \right|$$

by selecting a suitable $\Pi(\rho, \sigma)$. We can pull $|\sigma|$ at the denominator out of the absolute value and equivalently minimize

$$\sum_{s \in \sigma} \left| \Pi(\rho, \sigma) - \frac{|\sigma|}{|\rho|} \sum_{r \in \rho} P(r, s) \right|.$$

We recognize this as a problem of minimizing the sum of the absolute deviations, for which solution is known [29] to be

$$\operatorname*{med}_{s \in \sigma} \left\{ \frac{|\sigma|}{|\rho|} \sum_{r \in \rho} P(r, s) \right\} = \frac{|\sigma|}{|\rho|} \operatorname*{med}_{s \in \sigma} \left\{ \sum_{r \in \rho} P(r, s) \right\} = \Pi_{med}(\rho, \sigma).$$

As will be shown in the experimental evaluation, the median-based scheme results in the most accurate error bound. Note that in this case Π_{med} might not be stochastic.

3.4. Adaptive state-space aggregation procedure

Algorithm 1 demonstrates how the theoretical framework developed in the previous subsection is applied to the aggregating analysis of a DTMC. First, we construct a clustering Φ of the state space S with the use of the *partition* procedure (line 2), which will be described later in more detail. Then, we directly apply the corresponding definitions to compute abstract transient probabilities π_0 (see (2)), abstract transition probabilities Π (using either (17), (18) or (19)), error factors τ (see (15)) and the aggregation error $\|e_0\|_1$. We proceed by carrying out vector-matrix multiplications in the abstract setting and updating

Algorithm 1 Adaptive state-space aggregation of DTMC

```
    Input: DTMC $(S, P)$, initial probability distribution $p_0$, time horizon $k$.
    Output: Approximate probability distribution $\tilde{p}_k$, error bound $\|e_k\|_1$
 1: $\tilde{p}_0 \leftarrow p_0$
 2: $\Phi \leftarrow partition(S, P, \tilde{p}_0)$
 3: $(\pi_0, \Pi, \tau, aggregationError) \leftarrow aggregate(\Phi, P, \tilde{p}_0)$
 4: $error_0 \leftarrow aggregationError$
 5: for $i \leftarrow 1$ to $k$ do
 6:     $error_i \leftarrow error_{i-1} + \sum_{\rho \in \Phi} \pi_{i-1}(\rho) \cdot \tau(\rho)$
 7:     $\pi_i \leftarrow \pi_{i-1} \cdot \Pi$
 8:     if $checkPartition(\Phi, \pi_i)$ then
 9:         $\tilde{p}_i \leftarrow deaggregate(\pi_i, \Phi)$
10:         $\Phi \leftarrow partition(S, P, \tilde{p}_i)$
11:         $(\pi_i, \Pi, \tau, aggregationError) \leftarrow aggregate(\Phi, P, \tilde{p}_i)$
12:         $error_i \leftarrow error_i + aggregationError$
13: $\tilde{p}_k \leftarrow deaggregate(\pi_k, \Phi)$
14: return $\tilde{p}_k, error_k$
```

the error bound according to Equation (16). After several discrete steps, the probability distribution π_i might start to generate a large amount of error, which is signalised by the *checkPartition* procedure (line 8, also described later), and the re-partitioning of the state space takes place.

From the computational standpoint, lines 6 and 7 represent the critical parts of Algorithm 1 and are equivalent to computing a scalar product and a vector-matrix multiplication in the abstract setting, i.e. with the reduced state space. In the case where $|\Phi| \ll |S|$, we can achieve a considerable speedup over the computation on the concrete chain, provided that reaggregations are performed not too often (yet not too rarely in order to keep the accrued error relatively low). The remainder of this subsection describes the partitioning of arbitrary Markov chains (the *partition* procedure) as well as how often should reaggregations occur (the *checkPartition* procedure).

Assume that we have a DTMC $D = (S, P)$ having initial probability distribution p_0. We wish to construct a partition Φ of S that, for a current probability distribution, would provide an accurate and efficient approximation of D. We can view the DTMC D as a directed weighted graph, where nodes correspond to states and edges correspond to (non-zero) transition probabilities, and construct a partition Φ of S by clustering together adjacent nodes, i.e. states connected by a transition. Furthermore, inspect update Equation (16) for the error bound: the product of $\tau(\cdot)$ and $\pi_k(\cdot)$ represents an error contributed by a cluster, and we should minimize $\tau(\cdot)$ for clusters that bear a significant probability mass. While experimenting with various partitions, we have often noticed that clusters of large size are typically associated with large values of $\tau(\cdot)$. This makes sense: having (a large amount of) smaller and more refined clusters allows to track the probability mass more accurately than using few large clusters. Figure 2 illustrates the behaviour of both aggregation and propagation errors, with the latter being attributed to a phenomenon we call 'probability forwarding': when propagating probability from cluster ρ to cluster σ, we are effectively pushing the probability mass (in one step) to those states in σ that were previously unreachable (in one step) from any of the states in ρ; conversely, we are pushing the probability to cluster σ from those states in ρ that do not have any of the states in σ as their direct successor. By doing so, we effectively accelerate the model and incur in an error. Therefore, we can safely use clusters of big size for groups of states possessing a small probability mass: an insignificant $\pi_k(\cdot)$ will cancel large $\tau(\cdot)$; conversely, we should group states with significant probability into small clusters, possibly leaving such states as singletons.

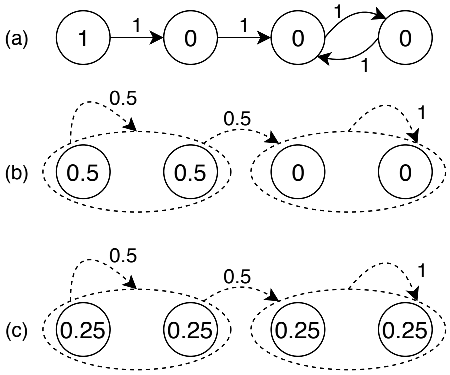

Figure 2: In (a) we consider a simple DTMC that deterministically starts in the leftmost state (numbers inside nodes denote current transient probabilities); in (b) we construct its aggregation (here we use outgoing averaging to compute abstract transitions), notice how the first (second) state has effectively lost (gained) some probability mass, resulting in an aggregation error; in (c) we perform one iteration in the abstract model, observe that the right-most state has effectively gained – due to 'probability forwarding' – some probability mass, resulting in a propagation error: in the concrete model this state is unreachable until the third iteration.

The partition with properties described above can be easily constructed in the process of hierarchical bottom-up clustering of the transition graph that represents the state space of the model. We make use of the nearest-neighbor approach, where we sequentially merge pairs of clusters corresponding to pairs of states having the closest 'distance' in the sense of mutual transition probabilities. During this process we prevent the creation of clusters with size exceeding a given parameter *maxClusterSize*. In the end, we obtain a prototype partition Φ, which we then use to derive – based on the current probability distribution – individual clusters Φ_j used for probability propagation. These partitions are obtained by splitting into singletons those clusters whose total probability exceeds a given *aggregation threshold* δ_{agg}. The resulting partition contains either singletons or composite clusters drawn from the prototype clustering Φ. This approach is the result of experiments (cf. relevant section), where we have tried to minimize the cost of reaggregation (in order to be able to reaggregate the model more frequently and provide a better approximation): the presented approach achieves this by pre-computing information in advance, primarily state-to-cluster and cluster-to-cluster data, which is used during the computation of the abstract transition probability matrix Π as well as of τ-factors.

Finally, as the model evolves and the probability distribution shifts, at some point large clusters may accumulate a significant portion of the probability mass and start producing a considerable amount of error. The *checkPartition* procedure detects this by testing whether partition Φ_j contains a composite cluster with a probability mass exceeding a predefined *reaggregation threshold* δ_{reagg}. After that, a new clustering Φ_{j+1} respecting current probability distribution is established.

The choice of parameters *maxClusterSize*, δ_{agg} and δ_{reagg} offers flexibility and affects the trade-off between efficiency and accuracy. In particular, small values of *maxClusterSize* will lead to a more refined prototype clustering, although the resulting aggregated state space can be large. Large values of aggregation threshold δ_{agg} will preserve a lot of clusters and lead to a greater state-space reduction at a price of decreased precision. Finally, a small value of reaggregation threshold δ_{reagg} usually implies frequent reaggregations and offers greater precision; it is recommended to set this parameter a couple of orders of magnitude larger than the aggregation threshold δ_{agg}.

4. State-Space Aggregation of CTMCs

Having established an aggregation method for DTMCs, let us now combine it with a uniformisation technique, in order to analyse continuous-time chains. We begin with SU, where the uniformisation rate is fixed and the application of aggregation described in the previous section is straightforward: the approach was already outlined in [10], although a rigorous formalisation is missing and is offered in this section. We then proceed by combining aggregation with AU, and, finally, we present a new hybrid method that combines the principles of both state-space aggregation and truncation, and offers an unprecedented flexibility for analysing continuous-time chains.

4.1. State-space aggregation with standard uniformisation (SU)

Recall that SU works by constructing a uniformised DTMC from the rate matrix using a single uniformisation rate q; it proceeds by computing transient probabilities for the DTMC and then weighs them using a Poisson distribution $\psi_{qt}(\cdot)$. The Fox-Glynn algorithm provides bounds L, R that allow to truncate the infinite sum and compute the overall result, as in (10). The error associated with this truncation is the probability loss $\|p_t\|_1 - \|\hat{p}_t\|_1 = 1 - \|\hat{p}_t\|_1$.

Now assume we use state-space aggregation to approximate transient probabilities u_k of the uniformised DTMC with the $\tilde{u}_k$ and obtain an approximation $\tilde{\hat{p}}_t$ of $\hat{p}_t$ as:

$$\tilde{\hat{p}}_t(s) \coloneqq \sum_{k=L}^{R} \psi_{qt}(k) \cdot \tilde{u}_k(s). \tag{20}$$

Each $\tilde{u}_k$ has an associated uncertainty $\|e_k\|_1$ and each $\tilde{u}_k$ is weighted with $\psi_{qt}(k)$, so the overall error is $\sum_{k=L}^{R} \|e_k\|_1 \cdot \psi_{qt}(k)$. We also lose some probability mass by truncating the infinite summation using bounds L and R, and to quantify this loss we use the following lemma.

Lemma 1. Let $\{v_k\}_{k\in\mathbb{N}_0}$ be an infinite series of vectors and let $v := \sum_{k=0}^{\infty} w_k \cdot v_k$, where w_k are non-negative scalars for which $\sum_{k=0}^{\infty} w_k = 1$. If norms of v_k are bounded by some v^*, i.e. $\exists v^* \; \forall k \in \mathbb{N}_0 \colon \|v_k\|_1 \leq v^*$, then $\|v\|_1 \leq v^*$.

Proof.

$$\|v\|_1 = \left\| \sum_{k=0}^{\infty} w_k \cdot u_k \right\|_1 \leq \sum_{k=0}^{\infty} w_k \cdot \|u_k\|_1 \leq \sum_{k=0}^{\infty} w_k \cdot v^* = v^* \sum_{k=0}^{\infty} w_k = v^*.$$

□

Corollary 1. Let $\hat{v} := \sum_{k=L}^{R} w_k \cdot v_k$ be a truncation of v for arbitrary bounds L, R, where $\{w_k\}$ and $\{v_k\}$ are consistent with Lemma 1. Then,

$$\|v\|_1 - \|\hat{v}\|_1 \leq v^* - \|\hat{v}\|_1 . \tag{21}$$

This corollary asserts that for bounded vectors u_k used in (10) (or for bounded approximations $\tilde{u}_k$ of u_k used in (20)), we are able to compute the probability loss resulting from the truncation of the infinite sum. For SU, AU, FSU or FAU, each of the $\|u_k\|_1$ is bounded by 1 and so probability loss in (21) takes the usual form $1 - \|\hat{p}_t\|_1$. However, for general state-space aggregations, there is no guarantee on the approximations $\tilde{u}_k$, because we do not require the transition matrices Π to be stochastic. Therefore, in order to safely use aggregation techniques in combination with the Fox-Glynn scheme, it is crucial to appeal to those abstractions for which $\{\tilde{u}_k\}$ is consistent with Lemma 1. An ideal candidate is the outgoing averaging approach (18) that preserves the stochasticity of vectors $\tilde{u}_k$, i.e. $\|\tilde{u}_k\|_1 = 1$, and so the probability loss can be computed analogously to SU/AU. As will be demonstrated on various case studies, outgoing averaging is the most efficient aggregating scheme for the analysis of discrete-time chains.

The rest of the algorithm remains conceptually the same: we work with the abstraction of the uniformised DTMC, where the notions of state adjacency, partition checking, etc. are analogous to those developed in the previous subsection. We compute approximate transient probability distributions of this DTMC that are weighted by Poisson probabilities. The overall error is the sum of the probability loss and an approximation error for the uniformised DTMC, namely:

$$\|e_t\|_1 \leq 1 - \left\| \tilde{\hat{p}}_t \right\|_1 + \sum_{k=L}^{R} \|e_k\|_1 \cdot \psi_{qt}(k). \tag{22}$$

The combination of the state-space aggregation with SU is denoted SU+.

4.2. State-space aggregation with adaptive uniformisation (AU)

As mentioned earlier, introducing adaptivity to the uniformisation procedure can greatly reduce the required number of discrete steps without any penalty in precision. We wish to incorporate this principle to the aggregating method described above. We cannot apply adaptive uniformisation directly since we have developed the aggregation scheme for (uniformised) DTMCs: we would have to continuously recompute its transition probability matrix each time a uniformisation rate q_i changes, which is impractical. Instead, this matrix can be defined as a function of the uniformisation rate (q), as shown next.

Let (S, R) be a CTMC and Q be the infinitesimal generator associated with R. Assume a specific state-space aggregation Φ of S is given. Using outgoing averaging (18), we have

$$\Pi(\rho, \sigma) = \frac{1}{|\rho|} \sum_{r \in \rho} \sum_{s \in \sigma} unif_R^q(r, s), \tag{23}$$

where q is the uniformisation rate. In the case where $\rho \neq \sigma$, we have $\forall r \in \rho$ $\forall s \in \sigma \colon r \neq s$, so that $unif_R^q(r, s) = \frac{Q(r,s)}{q}$ and therefore Equation (23) becomes

$$\Pi(\rho, \sigma) = \frac{1}{|\rho|} \sum_{r \in \rho} \sum_{s \in \sigma} \frac{Q(r, s)}{q} = \frac{1}{q} \cdot \frac{1}{|\rho|} \sum_{r \in \rho} \sum_{s \in \sigma} Q(r, s).$$

On the other hand, if $\rho = \sigma$, we obtain:

$$\begin{aligned} \Pi(\rho, \rho) &= \frac{1}{|\rho|} \sum_{r \in \rho} \left[1 + \frac{Q(r, r)}{q} + \sum_{s \in \rho, r \neq s} \frac{Q(r, s)}{q} \right] = \frac{1}{|\rho|} \sum_{r \in \rho} \left[1 + \sum_{s \in \rho} \frac{Q(r, s)}{q} \right] \\ &= \frac{1}{|\rho|} \left[\sum_{r \in \rho} 1 + \sum_{r \in \rho} \sum_{s \in \rho} \frac{Q(r, s)}{q} \right] = 1 + \frac{1}{q} \cdot \frac{1}{|\rho|} \sum_{r \in \rho} \sum_{s \in \rho} Q(r, s). \end{aligned}$$

We arrive at the following definition:

Definition 6. Let (S, R) be a CTMC with infinitesimal generator Q and let Φ be the partition of S. An abstract infinitesimal generator $\Theta \colon \Phi \times \Phi \to \mathbb{R}$ is the matrix defined as

$$\Theta(\rho, \sigma) = \frac{1}{|\rho|} \sum_{r \in \rho} \sum_{s \in \rho} Q(r, s).$$

Furthermore, the exit rate of the abstract state σ is the maximum exit rate of states within this cluster: $E(\sigma) \coloneqq \max_{s \in \sigma} E(s)$.

Observe the similarity between this definition and Equation (18) defining the abstract transition probability matrix based on outgoing probability averaging. Given this definition, we tailor the derivation above to the following results, of which only the Corollary requires details of proof.

Proposition 1. Let (S, R) be a CTMC, Φ be the partition of S, Θ be the corresponding abstract infinitesimal generator and $q \geq \max_{\sigma \in \Phi} E(\sigma)$ be the uniformisation rate. Then for the abstract transition probability function Π based on outgoing averaging of its uniformised DTMC it holds:

$$\Pi(\rho, \sigma) = \begin{cases} 1 + \frac{\Theta(\rho,\sigma)}{q}, & \text{if } \rho = \sigma \\ \frac{\Theta(\rho,\sigma)}{q}, & \text{otherwise.} \end{cases}$$

Corollary 2. For the error factors $\tau(\cdot)$ associated with partition Φ and abstract transition function Π based on average outgoing probabilities, the following holds (notice how this expression mirrors Equation (15) derived for the error factors in the discrete-time case):

$$\tau(\rho, \sigma) = \frac{1}{q} \sum_{s \in \sigma} \left| \frac{\Theta(\rho, \sigma)}{|\sigma|} - \frac{1}{|\rho|} \sum_{r \in \rho} Q(r, s) \right|.$$

Proof. Consider the case $\rho \neq \sigma$. Starting from Equation (15), we have:

$$\begin{aligned} \tau(\rho, \sigma) &= \sum_{s \in \sigma} \left| \frac{\Pi(\rho, \sigma)}{|\sigma|} - \frac{1}{|\rho|} \sum_{r \in \rho} unif_R^q(r, s) \right| \\ &= \sum_{s \in \sigma} \left| \frac{1}{|\sigma|} \frac{\Theta(\rho, \sigma)}{q} - \frac{1}{|\rho|} \sum_{r \in \rho} \frac{Q(r, s)}{q} \right| = \frac{1}{q} \sum_{s \in \sigma} \left| \frac{\Theta(\rho, \sigma)}{|\sigma|} - \frac{1}{|\rho|} \sum_{r \in \rho} Q(r, s) \right|. \end{aligned}$$

If instead $\rho = \sigma$, we have:

$$\begin{aligned} \tau(\rho, \sigma) &= \sum_{s \in \sigma} \left| \frac{\Pi(\rho, \sigma)}{|\sigma|} - \frac{1}{|\rho|} \sum_{r \in \rho} unif_R^q(r, s) \right| \\ &= \sum_{s \in \sigma} \left| \frac{1}{|\sigma|} \left[1 + \frac{\Theta(\rho, \sigma)}{q} \right] - \frac{1}{|\rho|} \left[1 + \frac{Q(r, r)}{q} + \sum_{s \in \sigma, r \neq s} \frac{Q(r, s)}{q} \right] \right| \\ &= \sum_{s \in \sigma} \left| \frac{1}{|\sigma|} + \frac{1}{|\sigma|} \frac{\Theta(\rho, \sigma)}{q} - \frac{1}{|\rho|} - \frac{1}{|\rho|} \sum_{r \in \rho} \frac{Q(r, s)}{q} \right| \\ &= \frac{1}{q} \sum_{s \in \sigma} \left| \frac{\Theta(\rho, \sigma)}{|\sigma|} - \frac{1}{|\rho|} \sum_{r \in \rho} Q(r, s) \right|. \end{aligned}$$

□

The algebraic manipulations above have expressed $\Pi(\cdot, \cdot)$ and $\tau(\cdot)$ in terms of the uniformisation rate q. When this rate varies at each iteration, we are able to construct $\Pi(\cdot, \cdot)$ for the probability propagation, as well as $\tau(\cdot)$ for computing the error bound, on the fly. We will no longer work with P and its abstraction Π, but with Q and its abstraction Θ: the structure (Φ, Θ) is *an abstract CTMC*.

Using the definition of uniformisation rate (11) along with the Definition 6 for the exit rate of a cluster, we can compute the current uniformisation rate q_i as

$$q_i \geq \max_{s \in S}\{E(s) \mid \tilde{u}_i(s) > 0\} = \max_{\sigma \in \Phi}\{E(\sigma) \mid \pi_i(\sigma) > 0\}.$$

The resulting method, denoted as AU+, integrates state-space aggregation with adaptive uniformisation and its procedure can be summarised as follows. First, we construct a prototype partition Φ of S in the same way as for discrete-time chains. Then we construct a concrete clustering Φ_0 by splitting into singletons abstract states with probability higher than the specified aggregation threshold δ_{agg}. Afterwards, we establish abstract infinitesimal generator Θ_0 corresponding to partition Φ_0, compute for each cluster σ the exit rate according to Definition 6 and the value of τ:

$$\sum_{\sigma \in \Phi} \sum_{s \in \sigma} \left| \frac{\Theta(\rho, \sigma)}{|\sigma|} - \frac{1}{|\rho|} \sum_{r \in \rho} Q(r, s) \right|,$$

which, according to Corollary 2, when divided by the current (local) uniformisation rate q_i, will yield the value of $\tau(\rho)$. We have obtained an aggregated abstract CTMC, which is a normal CTMC (albeit with special exit rates) and can be passed as an input to the adaptive uniformisation procedure. AU unfolds the model while the partition checker ensures the correctness of the given partition and triggers reclusterings when necessary. The final transition probability as well as the approximation error are computed similarly to (20) and (22) in the case of SU, but using the solution of the birth process as a weighting factor.

4.3. Combination of state-space aggregation with truncation

Finally, we can combine the method above with the state-space truncation where in each iteration we truncate insignificant clusters, i.e. those with transient probability smaller than the truncation threshold δ_{tru}, in order to obtain even smaller uniformisation rates. The resulting technique, denoted FAU+, incorporates both state-space aggregation and state-space truncation. The aggregation threshold δ_{agg} that defines how much of the probability mass can constitute a cluster, along with the truncation threshold δ_{tru} defining which abstract states are to be considered insignificant, drive the overall behaviour of the method and offer a great flexibility for handling various types of models, as will be shown later. On a final note, notice that, when setting aggregation and truncation thresholds, we can categorise the approximation techniques as in Table 1.

5. Experimental Evaluation

The goal of this section is to present an exhaustive experimental evaluation of approximate methods for the analysis of DTMCs and CTMCs. We will start in a discrete-time setting, examine the behaviour of various aggregation

δ_{agg}	δ_{tru}	Resulting method
$= 0$	$= 0$	AU
$= 0$	> 0	FAU
> 0	$= 0$	AU+
> 0	> 0	FAU+

Table 1: Categorisation of the approximation techniques.

schemes, evaluate the quality of the theoretical bounds, and finally compare the best aggregating strategy against a method based on state-space truncation. Finally, we will explore how these methods – including the novel hybrid approach (FAU+) – compare in the continuous-time setting.

5.1. Experimental setup

All methods are implemented in PRISM [7], a state-of-the-art probabilistic model-checker. We employ its explicit engine, which provides the best performance for models of moderate size (up to $\approx 10^7$ states) that do not exhibit regular or symmetric structures [7]. All the experiments are run on a Debian server with 8x Intel Core i7-3770K CPUs (4 cores at 3.5 GHz) and 32 GB RAM, with all the algorithms being executed sequentially (1 thread).

The benchmark comprises four models (originally all in continuous time) from different application areas: the Lotka-Volterra model [23], a prokaryotic gene expression [30] model, the model of a workstation cluster [31], and the model of a two-component signalling pathway [32]. For the sake of generality and fairness, these models have been chosen to cover the broad range of possible behaviours that a stochastic process can exhibit. For instance, the signalling pathway is characterized by its wide spread of probability distribution, which limits the reduction capabilities of approximate methods. On the other hand, the workstation cluster model exhibits complex dynamics for the transient probability propagation. The Lotka-Volterra model, despite being simple and predictable, has fast dynamics. We will discuss how these differences impact the behaviour of individual approximate techniques, where aggregation-based methods handle such models differently than those based on state-space truncation.

5.2. Aggregation of DTMCs

5.2.1. Precision of different aggregation strategies

In our first set of experiments, we will inspect the behaviour of individual aggregation schemes in various scenarios for a DTMC. We are interested in the obtained precision (or accuracy), both empirical (actual distance from the exact DTMC solution) and theoretical (derived upper bounds on the L_1-norm of the error vector). To eliminate any model bias, we perform experiments on two different models with different sizes and exhibiting distinctive behaviours (see Tables 2 and 3). We perform three different experiments, having the following separate goals:

- Experiment 1 (with outcomes **E1(e)**, **E1(t)**): The goal of this experiment is to compare a one-step behaviour of various abstract transition functions. We evolve the model for 100 (concrete) steps, then perform the first partitioning and compute abstract transition matrices using three different approaches: median-based (Med), based on average incoming (In and In’) or outgoing (Out) transition probabilities. Then we perform a single step in the abstract model. In all cases we use the same value of aggregation threshold and therefore each scheme will be working with exactly the same state space partition. We report empirical error (**E1(e)**) and theoretical error bound on the propagation error (**E1(t)**), i.e. $\|e_{101}\|_1 - \|e_{100}\|_1$ (that is, we do not take aggregation error $\|e_{100}\|_1$ into account, as it would be same in all cases since each technique uses the same state-space partition). The differences in the obtained values arise only from using different abstract transition matrices. For the case of average incoming probabilities, we also compute theoretical bound using the ϵ-factors, as in (6) (In’), and the new bound (16) based on τ-factors (In).

- Experiment 2 (**E2(e)**, **E2(t)**): The goal of this experiment is to demonstrate the long-term behaviour of different aggregation approaches. It is the same as **E1(e)** and **E1(t)**, but after the initial aggregation we perform 100 consecutive steps without reclusterings. The value reported in **E2(t)** is $\|e_{200}\|_1 - \|e_{100}\|_1$, i.e. the bound on the propagation error 100 steps after the first aggregation.

- Experiment 3 (**E3(e)**, **E3(t)**): The goal of this experiment is to investigate the behaviour of various approximations under regular re-aggregations. It is the same as **E2(e)** and **E2(t)**, but in the course of 100 steps after the first aggregation we perform 10 additional reclusterings at fixed times (after iteration 105, 115, 125 etc.). Since for each of the schemes the probability distributions during the corresponding times will be approximately the same and the aggregation error is negligible compared to the propagation one, in **E3(t)** we report only the bound on the propagation error.

	Med	In’	In	Out
E1(e)	**1.93E-23**	**2.07E-23**	**2.07E-23**	2.51E-23
E1(t)	**9.77E-24**	1.31E-20	**1.28E-23**	1.65E-23
E2(e)	**3.59E-4**	**3.58E-4**	**3.58E-4**	6.40E-4
E2(t)	**3.71E-4**	1.33E-1	**3.86E-4**	8.95E-4
E3(e)	3.50E-4	2.01E-17	2.01E-17	**2.85E-20**
E3(t)	3.50E-4	7.17E-14	2.81E-17	**2.04E-19**

Table 2: Precision of different aggregation schemes. Model: Lotka-Volterra, dimension of the state space: 160k, aggregation threshold: 1E-25.

	Med	In’	In	Out
E1(e)	**1.24E-9**	1.26E-9	1.26E-9	1.29E-9
E1(t)	**8.09E-11**	4.39E-6	1.07E-10	1.27E-10
E2(e)	**1.25E-7**	**1.21E-7**	**1.21E-7**	1.94E-7
E2(t)	**1.40E-7**	1.06E-2	**1.52E-7**	2.80E-7
E3(e)	1.27E-7	1.02E-7	1.02E-7	**1.68E-8**
E3(t)	1.27E-7	2.10E-4	1.11E-7	**3.17E-8**

Table 3: Precision of different aggregation schemes. Model: Prokaryotic Gene Expression, dimension of the state space: 700k, aggregation threshold: 1E-10.

The results for two different models (with diferrent state-space sizes) are shown in Tables 2 and 3. First, from all experiments we confirm that In $\ll$ In’, i.e. the newly derived error bound (16) that utilises τ-terms provides several orders of magnitude better bounds than that based on ϵ-terms (6). Second, Experiment 1 shows us that median-based aggregation exhibits the best one-step behaviour, followed by that based on incoming probability, and finally that based on outgoing probability. Third, in Experiment 2 we see a clear decrease in precision over all the methods: the probability distribution has changed and in the absence of reclusterings we obtain a significant error. Fourth, although median-based aggregation performs slightly worse than incoming averaging, its theoretical bound **E2(t)** on the actual error is still the best. Finally, in Experiment 3 we see that reclusterings can drastically improve the speedup of the approaches based on incoming and outgoing probability, with the latter improving of a couple of orders of magnitude.

Remark. The difference in the obtained values arises from how individual schemes handle the problem of probability forwarding into big clusters. Median-based aggregation is very likely to select the median transition probability between clusters to be equal to zero, because a majority of states in a big target cluster would be inaccessible in one step. Hence, no probability forwarding occurs at all, and the error is generated by the opposite effect: states that are accessible in one step will not get any probability at all. In the long run, it seems to be ineffective because zero abstract transitions decelerate the model and reaggregations cannot improve this; median-based aggregation, as expected, produces the best error bound, yet this empirical error is poor. In fact, during the evaluation of large models (e.g. in experiments presented later), the median-based strategy has failed to produce any reasonable results. On the other hand, strategies based on averaging always propagate at least some probability mass, and the incoming version seems to be advantageous because a large size of a successor can alleviate abstract transition probability and probability forwarding is less apparent as compared to outgoing. The latter, however, seems to yield more precise resuls under regular reclusterings.

5.2.2. Speedup of approximate methods

In the next set of experiments we investigate the speedup of individual approximation methods. The choice of the model is arbitrary, in the sence that, in principle, none of the methods are more advantageous than others. We select a Lotka-Volterra model of 0.5M states, and compute an approximation of its transient probability distribution at time 10000. For a given precision (Prec), ranging from 1e-1 to 1e-5, each method is tasked with computing as fast as possible while guaranteeing this precision; the precision of each method is computed using τ-factors (16) for outgoing (Out) and incoming (In) averaging, using ϵ-factors (6) for incoming (In') averaging, and the using probability loss for the state-space truncation (Tru). To ensure a fair comparison, parameters for each of the methods are tuned individually in each of the experiments in order to obtain the best computation time. Results of this experiment are presented in Table 4 where we report the speedup with respect to a reference computation (10000 usual multiplications of matrices of size 0.5M).

Precision	Tru	In'	In	Out
1E-5	**4.541**	2.650	3.181	**4.715**
1E-4	**4.541**	2.785	3.201	**4.789**
1E-3	**4.720**	2.785	3.201	**5.206**
1E-2	**4.944**	2.935	4.059	**5.712**
1E-1	**5.142**	3.380	4.059	**6.029**

Table 4: Speedup (acceleration w.r.t. concrete computation for guaranteeing a given precision) of various approximate techniques. Model: Lotka-Volterra, dimension of the state space: 0.5M states, discrete time horizon: 10000 steps.

From Table 4, it is clear that Out > In > In'. The second inequality is due to the usage of better theoretical bounds, which means that incoming averaging that utilised τ instead of ϵ can allow for a larger empirical error (by clustering more or reclustering less) to guarantee a certain precision. The first inequality can be explained by the fact that, as we have seen previously, outgoing averaging is more susceptible to reaggregations, and therefore fewer of those are required to guarantee a certain precision. We thus confirm the results from the first set of experiments. Also, we know that outgoing averaging is the only strategy that can be safely used for the analysis of CTMCs, so, from now on, under adaptive aggregation we will mean the aggregation based on outgoing averaging that utilises τ-terms to quantify the theoretical error - we have now repeatedly seen that this approach that preserves the properties of DTMCs indeed exhibits the best behaviour.

What is new here is an illustration of a behaviour of the state space truncation Tru, which seems to be inferior to outgoing averaging. To eliminate any bias, let us further investigate this relationship by repeating the experiment with different models of larger size and with different dynamics. As previously, we investigate the speedup (acceleration w.r.t. the exact computation) for aggregation (Agg) and truncation (Tru), with both guaranteeing a given precision

(Prec). Tables 5, 6 and 7 present the results for the workstation cluster model, the uniformised prokaryotic gene expression model and the uniformised two-component signalling pathway, respectively. Also, in the third case, instead of transient analysis, both techniques are applied for a model checking problem over the Markov chain model.

Precision	Tru	Agg
1E-7	6.66	**9.10**
1E-6	7.55	**10.56**
1E-5	8.28	**11.21**
1E-4	8.86	**16.55**
1E-3	9.78	**17.96**

Table 5: Speedup (acceleration w.r.t. concrete computation for guaranteeing a given precision) comparison. Model: workstation cluster, dimension of the state space: 1M, time horizon: 10804.

Precision	Tru	Agg
1E-7	7.02	**21.71**
1E-6	9.16	**28.40**
1E-5	9.86	**35.78**
1E-4	10.97	**40.16**
1E-3	11.81	**51.00**

Table 6: Speedup (acceleration w.r.t. concrete computation for guaranteeing a given precision) comparison. Model: uniformised prokaryotic gene expression; dimension of the state space: 1.2M, time horizon: 10000.

Prec	Tru	Agg
1E-7	4.81	**11.54**
1E-6	5.76	**14.97**
1E-5	6.54	**16.75**
1E-4	7.37	**18.74**
1E-3	8.46	**24.13**

Table 7: Speedup (acceleration w.r.t. concrete computation for guaranteeing a given precision) comparison. Model: uniformised two-component signalling pathway, population bounds [14,46]; property of interest is $\mathbb{P}(\Box^{\leq 10000} popRp \leq 27)$; number of states after PCTL driven transformation: 949376.

In all cases the state-space aggregation has performed considerably better than truncation, demonstrating a two- to five-fold acceleration. In order to further investigate why this is the case, let us run a similar experiment as the one presented in Table 2, where we investigate the precision of the methods. We select the same Lotka-Volterra model of size 160k and evaluate it using aggregation (Agg) and truncation (Tru). The experiment goes as follows. First, we perform 100 exact steps, then we start approximating using the same aggregation/truncation threshold: with aggregation, states with probability below this threshold are aggregated; with truncation, such states are truncated. This way we perform 1, 100, 300, 500, 700 or 900 discrete steps more (in the case of aggregation, we also perform regular reclusterings, again at fixed times after 10 steps) and in each case we report empirical error (e) and theoretical (t) bound for both methods (for state-space truncation both errors are equal to the probability loss). The results are presented in Table 8. Recall that, contrary to experiments from Tables 4, 5 and 6, both aggregation and truncation use the same threshold.

Steps	Tru	Agg(e)	Agg(t)
101	1.0E-8	1.2E-8	1.4E-8
200	5.2E-7	2.1E-7	2.3E-6
400	2.0E-6	2.6E-7	7.7E-6
600	4.1E-6	3.1E-7	1.5E-5
800	6.2E-6	3.2E-7	2.0E-5
1000	7.9E-6	2.1E-7	2.4E-5

Table 8: Adaptive aggregation versus state space truncation precision comparison. Model: Lotka-Volterra, dimension of the state space: 160k, aggregation/truncation threshold: 1E-25.

One can see that aggregation gives a better empirical error, which confirms the intuition that aggregating the state space and having at least an approximate idea where the residual probability is located is better than truncating the state space. On the other hand, truncation-based method provide an excellent theoretical bound on the error that ultimately beats the approximation bounds based on τ-terms. However, as Tables 4, 5 and 6 suggest, this does not give truncation an advantage: when allowed to tune parameters individually, outgoing averaging is capable of striking a perfect balance between state-space reduction and number of reclusterings, in order to provide a significantly more efficient approximation.

5.3. Approximation of CTMCs

Now we wish to investigate both uniformisation techniques utilising state-space truncation (FSU & FAU) and combinations of SU & FAU with the state-space aggregation (SU+ & FAU+). The comparison of precision for aggregation versus truncation from the discrete-time case (Table 8) automatically translates to SU+ versus FSU, since now we compute the same transient probabilities of the uniformised model and only weigh them with Poisson probabilities afterwards. The precision comparison involving FAU or FAU+ is trickier instead, because of varying uniformisation rates. However, AU differs from SU only in the total number of iterations and not in the overall precision, so the conclusions from Table 8 can also be applied when comparing SU+/FAU+ with FAU.

We evaluate the overall speedup of individual techniques on various case studies, similarly to Tables 4 through 6. In Tables 9 through 11, for each of the techniques we report the time acceleration (with respect to SU) to guarantee a given precision (Prec).

Precision	FSU	FAU	SU+	FAU+
1E-7	4.96	5.06	**9.88**	**9.98**
1E-6	5.34	5.75	**11.61**	**12.18**
1E-5	5.20	6.08	**11.61**	**14.94**
1E-4	5.65	8.66	**17.85**	**20.18**
1E-3	5.79	8.67	**21.70**	**29.58**

Table 9: Speedup (acceleration w.r.t. SU for guaranteeing a given precision) comparison. Model: workstation cluster [31], dimension of the state space: 1M, upper Fox & Glynn bound: 10804.

Prec	FSU	FAU	SU+	FAU+
1E-7	4.95	6.17	**11.28**	**12.54**
1E-6	5.07	6.63	**12.16**	**13.77**
1E-5	5.07	7.02	**12.64**	**15.38**
1E-4	5.22	7.31	**13.72**	**16.52**
1E-3	5.38	7.58	**14.48**	**18.26**

Table 10: Speedup (acceleration w.r.t. SU for guaranteeing a given precision) comparison. Model: Lotka-Volterra, dimension of the state space: 1M, upper FG bound: 10117.

Prec	FSU	FAU	SU+	FAU+
1E-7	3.23	**5.55**	4.69	**5.00**
1E-6	3.66	**5.95**	5.54	**6.13**
1E-5	4.31	**6.80**	6.54	**7.29**
1E-4	5.08	**7.81**	7.45	**8.93**
1E-3	6.07	9.08	**9.12**	**11.86**

Table 11: Speedup (acceleration w.r.t. SU for guaranteeing a given precision) comparison. Model: signalling pathway, dimension of the state space: 1M, upper FG bound: 11013.

First, we observe that FSU is, without a doubt, inferior to both FAU and SU+. This is consistent with our intuition: FAU is capable to perform fewer discrete steps (approximately twice fewer) by varying the uniformisation rate; SU+ is better than FSU due to the same reasons why aggregation is generally favored against truncation for the DTMC analysis. Next, comparing FAU and SU+, we notice that for workstation cluster and Lotka-Volterra models aggregation is clearly the preferable approach, while for the signalling pathway FAU slightly outperforms aggregation. This example showcases how the model dynamics can influence the behaviour of the approximation method. While investigating the differences between these models, we have observed that in the first two cases – workstation cluster and Lotka-Volterra – a relatively small set of states maintains the majority of the probability mass, as compared to the signalling pathway where the probability distribution is "flatter". This difference implies that the state space of the first two models can be reduced much more, compared to the signalling pathway case (approximately 50x against 10x) and, as we can see, aggregation can benefit from this reduction much more. This may be attributed to the fact that the aggregation method initially operates on clusters (and there are clearly fewer of those of concrete states), which are then being adaptively deaggregated into singletons during the analysis. FAU, on the other hand, is capable of 'discovering' a newly active state (as opposed to newly active clusters) during probability transition, which, intuitively, seems to be more precise. However, for a small set of active states, as in the case of the workstation cluster of a Lotka-Volterra model, working with clusters to make use of the greater state space reduction (and then reaggregate more frequently to compensate for a minor precision loss) gives SU+ a two-fold advantage over FAU.

With regards to the hybrid FAU+ method, we can see that, in general, it manages to achieve balance between aggregation and truncation and, as a result, it noticeably outperforms standalone methods. However, this flexibility comes with a minor (but easily fixable) downside: the algorithm for this method is parametrized by variables (encompassing thresholds) associated with both aggregation and truncation, and it may not be apparent how to combine them to analyse a given method. We suggest that the aggregation threshold should be larger than the truncation one: we first aggregate states into clusters based on the first parameter and then truncate those abstract states according to the second one. Indeed, during the experimental evaluation of the method we have found that an optimal solution is to set the truncation threshold to be one to two orders of magnitude smaller.

6. Conclusions

The experimental evidence leads to several conclusions. First, we have explored various aggregation schemes and shown that the newly derived theoretical bound that utilises τ-factors provides a significant precision improvement over the existing bounds based on ϵ-terms. As a result, we have obtained an aggregation method that significantly outperforms truncation-based approach in the discrete case. Equipped with this method, we have designed an analogous aggregation technique for CTMCs and shown that it can provide an efficient approximation, both for transient analysis and for model checking problems. Second, we have investigated approaches based on aggregation and truncation, have observed how the model dynamics impacts the behaviour of the approximate methods, and have managed to distinguish classes of models for which aggregation-based methods are more appropriate than truncation-based, and vice-versa. Finally, we have succeeded to integrate both approaches into a new hybrid method (FAU+): its inherent flexibility handles models of arbitrary dynamics and, as a result, outmatches all existing methods.

References

[1] C. Baier, E. M. Hanh, B. R. Haverkort, H. Hermanns, J.-P. Katoen, Model checking for performability, Mathematical Structures in Computer Science 23 (4) (2013) 751–795.

[2] G. Bolch, S. Greiner, H. de Meer, K. S. Trivedi, Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications, Wiley-Interscience, New York, NY, USA, 1998.

[3] C. Madsen, C. J. Myers, N. Roehner, C. Winstead, Z. Zhang, Utilizing stochastic model checking to analyze genetic circuits, in: Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), IEEE, 2012, pp. 379–386.

[4] L. Cardelli, M. Kwiatkowska, M. Whitby, Chemical reaction network designs for asynchronous logic circuits, in: DNA Computing and Molecular Programming, Springer, 2016, pp. 67–81.

[5] J. Hey, R. Nielsen, Integration within the Felsenstein equation for improved Markov chain Monte Carlo methods in population genetics, Proceedings of the National Academy of Sciences 104 (8) (2007) 2785–2790.

[6] F. Didier, T. A. Henzinger, M. Mateescu, V. Wolf, Fast adaptive uniformization of the chemical master equation, in: High Performance Computational Systems Biology, 2009, pp. 118–127.

[7] M. Kwiatkowska, G. Norman, D. Parker, PRISM 4.0: Verification of probabilistic real-time systems, in: Computer Aided Verification (CAV), Springer, 2011, pp. 585–591.

[8] C. Dehnert, S. Junges, J.-P. Katoen, M. Volk, A Storm is coming: A modern probabilistic model checker, in: Computer Aided Verification (CAV), Springer, 2017, pp. 592–600.

[9] A. Hartmanns, H. Hermanns, The Modest Toolset: An integrated environment for quantitative modelling and verification, in: Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Springer, 2014, pp. 593–598.

[10] A. Abate, L. Brim, M. Češka, M. Kwiatkowska, Adaptive aggregation of Markov chains: Quantitative analysis of chemical reaction networks, in: Computer Aided Verification (CAV), Springer, 2015, pp. 195–213.

[11] A. P. A. van Moorsel, W. H. Sanders, Adaptive uniformization, Communications in Statistics. Stochastic Models 10 (3) (1994) 619–647.

[12] K. G. Larsen, A. Skou, Bisimulation through probabilistic testing, Information and Computation 94 (1) (1991) 1–28.

[13] J. Desharnais, F. Laviolette, M. Tracol, Approximate analysis of probabilistic processes: Logic, simulation and games, in: Quantitative Evaluation of Systems (QEST), 2008, pp. 264–273.

[14] A. D'Innocenzo, A. Abate, J.-P. Katoen, Robust PCTL model checking, in: Hybrid Systems: Computation and Control (HSCC), ACM, 2012, pp. 275–285.

[15] A. Abate, J.-P. Katoen, J. Lygeros, M. Prandini, Approximate model checking of stochastic hybrid systems, European Journal of Control 16 (6) (2010) 624 – 641.

[16] S. E. Z. Soudjani, A. Abate, Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes, SIAM Journal on Applied Dynamical Systems 12 (2) (2013) 921–956.

[17] S. E. Z. Soudjani, A. Abate, Precise approximations of the probability distribution of a Markov process in time: an application to probabilistic invariance, in: Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Springer, 2014, pp. 547–561.

[18] V. Chellaboina, S. P. Bhat, W. M. Haddad, D. S. Bernstein, Modeling and analysis of mass-action kinetics, IEEE Control Systems Magazine 29 (4) (2009) 60–78.

[19] L. Cardelli, M. Tribastone, M. Tschaikowski, A. Vandin, Maximal aggregation of polynomial dynamical systems, Proceedings of the National Academy of Sciences 114 (38) (2017) 10029–10034.

[20] J. Hasenauer, V. Wolf, A. Kazeroonian, F. Theis, Method of conditional moments (MCM) for the chemical master equation, Journal of Mathematical Biology 69 (3) (2014) 687–735.

[21] L. Cardelli, M. Kwiatkowska, L. Laurenti, A stochastic hybrid approximation for chemical kinetics based on the linear noise approximation, in: Computational Methods in Systems Biology (CMSB), Springer, 2016, pp. 147–167.

[22] M. Češka, J. Křetínský, Semi-quantitative abstraction and analysis of chemical reaction networks, in: Computer Aided Verification (CAV), Springer, 2019, pp. 475–496.

[23] D. T. Gillespie, Exact stochastic simulation of coupled chemical reactions, The journal of Physical Chemistry 81 (25) (1977) 2340–2361.

[24] J. Goutsias, Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems, The Journal of Chemical Physics 122 (18) (2005) 184102.

[25] H. Salis, Y. Kaznessis, Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions, The Journal of Chemical Physics 122 (5) (2005) 054103.

[26] Y. Cao, D. T. Gillespie, L. R. Petzold, The slow-scale stochastic simulation algorithm, The Journal of Chemical Physics 122 (1) (2005) 014116.

[27] B. Hepp, A. Gupta, M. Khammash, Adaptive hybrid simulations for multiscale stochastic reaction networks, The Journal of Chemical Physics 142 (3) (2015) 034118.

[28] B. L. Fox, P. W. Glynn, Computing Poisson probabilities, Communications of the ACM (CACM) 31 (4) (1988) 440–445.

[29] N. C. Schwertman, A. J. Gilks, J. Cameron, A simple noncalculus proof that the median minimizes the sum of the absolute deviations, The American Statistician 44 (1) (1990) 38–39.

[30] A. M. Kierzek, J. Zaim, P. Zielenkiewicz, The effect of transcription and translation initiation frequencies on the stochastic fluctuations in prokaryotic gene expression, The Journal of Biological Chemistry 276 (11) (2001) 8165–8172.

[31] B. Haverkort, H. Hermanns, J.-P. Katoen, On the use of model checking techniques for dependability evaluation, in: Reliable Distributed Systems, IEEE, 2000, pp. 228–237.

[32] M. Češka, D. Šafránek, S. Dražan, L. Brim, Robustness analysis of stochastic biochemical systems, PLOS ONE 9 (4) (2014) 1–23.