Semi-quantitative Abstraction and Analysis of Chemical Reaction Networks

Milan Češka[1(✉)] and Jan Křetínský[2]

[1] Brno University of Technology, FIT,
IT4I Centre of Excellence, Brno, Czech Republic
`ceskam@fit.vutbr.cz`
[2] Technical University of Munich, Munich, Germany

Abstract. Analysis of large continuous-time stochastic systems is a computationally intensive task. In this work we focus on population models arising from chemical reaction networks (CRNs), which play a fundamental role in analysis and design of biochemical systems. Many relevant CRNs are particularly challenging for existing techniques due to complex dynamics including stochasticity, stiffness or multimodal population distributions. We propose a novel approach allowing not only to predict, but also to explain both the transient and steady-state behaviour. It focuses on qualitative description of the behaviour and aims at quantitative precision only in orders of magnitude. First we build a compact understandable model, which we then crudely analyse. As demonstrated on complex CRNs from literature, our approach reproduces the known results, but in contrast to the state-of-the-art methods, it runs with virtually no computational cost and thus offers unprecedented scalability.

1 Introduction

Chemical Reaction Networks (CRNs) are a versatile language widely used for *modelling and analysis* of biochemical systems [12] as well as for high-level *programming* of molecular devices [8,40]. They provide a compact formalism equivalent to Petri nets [37], Vector Addition Systems (VAS) [29] and distributed population protocols [3]. Motivated by numerous potential applications ranging from system biology to synthetic biology, various techniques allowing simulation and formal analysis of CRNs have been proposed [2,9,21,24,39], and embodied in the design process of biochemical systems [20,25,32]. The time-evolution of CRNs is governed by the Chemical Master Equation (CME), which describes the probability of the molecular counts of each chemical species. Many important biochemical systems lead to complex dynamics that includes *state space explosion, stochasticity, stiffness, and multimodality* of the population distributions

[23,44], and that fundamentally limits the class of systems the existing techniques can effectively handle. More importantly, biologist and engineers often seek for plausible explanations why the system under study has or has not the required behaviour. In many cases, a set of system simulations/trajectories or population distributions is not sufficient and the ability to provide an accurate explanation for the temporal or steady-state behaviour is another major challenge for the existing techniques.

In order to cope with the computational complexity of the analysis and in order to obtain explanations of the behaviour, we shift the focus from quantitatively precise results to a more qualitative analysis, closer to how a human would behold the system. Yet we insist on providing at least rough timing information on the behaviour as well as rough classification of probability of different behaviours at the extent of "very likely", "few percent", "barely possible", so that we can conclude on issues such as time to extinction or bimodality of behaviour. This gives rise to our *semi-quantitative* approach. We stipulate that analyses in this framework reflect quantities in orders of magnitude, both for time duration and probabilities, but not more than that. This paradigm shift is reflected on two levels: (1) We abstract systems into semi-quantitative models. (2) We analyse systems in a semi-quantitative way. While each of the two can be combined with a traditional abstraction/analysis, when combined together they provide powerful means to understand systems' behaviour with virtually no computational cost.

Semi-quantitative Models. The states of the models contain information on the current amount of objects of each species as an interval spanning often several orders of magnitude, unless instructed otherwise. For instance, if an amount of a certain species is to be closely monitored (as a part of the input specification/property of the system) then this abstraction can be finer. Similarly, whenever the analysis of a previous version of the abstraction points to the lack of precision in certain states, preventing us to conclude which of the possible behaviours is prevalent, the corresponding refinement can take place. Further, the rates of the transitions are also captured only with such imprecision. The crucial point allowing for existence of such models that are small, yet faithful, is our concept of *acceleration*. It captures certain *sequences* of transitions. It eliminates most of the non-determinism that paralyses other types of abstractions, which are too over-approximative, unable to conclude anything, but safety properties.

Semi-quantitative Analysis. Instead of performing exact transient or steady-state analysis, we can consider most probable transitions and then carefully lift this to most probable temporal behaviours. Technically, this is done by *alternating between transient and steady-state analysis* where only some rates and transitions are taken into account at different stages. In order to further facilitate the resulting insight of the human on the result of the analysis, we provide an algorithm to perform this analysis with virtually no computation effort and thus possibly manually. The trivial computations immediately pinpoint why certain

behaviours occur. Moreover, less likely behaviours can also be identified easily, to any desired degree of improbability (dozens of percent, promilles etc.).

To summarise, the first step yields tiny models, allowing for a synoptic observation of the model; due to their size these models can be either analysed easily using standard means, or can be subject to the second step. The second step provides an efficient approximative analysis, which is also very illustrative due to the limited use of quantities. It can be applied to any system; however, it is particularly interesting in connection with the models coming from the first step since (i) no extra effort (size, computation) is wasted on overly precise treatment that is ignored by the other step, and (ii) together they yield an understandable explanation of the behaviour. An entertaining feature of this paradigm is that the stiffer (with rates at hugely different time scales) the system is the easier it is to analyse.

To demonstrate the capabilities of our approach, we consider three challenging and biologically relevant case studies that have been used in literature to evaluate state-of-the-art methods for the CRN analysis. It has been shown that many approaches fail, either due to time-outs or incapability to capture differences in behaviours, and some tailored ones require considerable computational effort, e.g. an hour of computation. Our experiments clearly show that the proposed approach can deliver results that yield qualitatively same information, more understanding and can be computed in minutes by hand (or within a fraction of a second by computer).

Our contribution can be summarized as follows:

– We propose a novel *semi-quantitative* framework for analysis of CRN and similar population models, focusing on explainability of the results and low complexity, with quantitative precision limited to orders of magnitude.
– An algorithm for abstracting CRNs into semi-quantitative models based on interval abstraction of the species population and on transition acceleration.
– An algorithm for semi-quantitative analysis that replaces exact numerical computation by exploring the most probable transitions and alternating transient and steady-state analysis.
– We consider three challenging CRNs thoroughly studied in literature and demonstrate that the semi-quantitative abstraction and analysis gives us a unique tool that is able to accurately predict and explain both transient and steady-state behaviour of complex CRNs in a fraction of a second.

Related Work

To the best of our knowledge, there does not exist any abstraction of CRNs similar to the proposed approach. Indeed, there exist various abstraction and approximation schemes for CRNs that improve the performance and scalability of both the simulation-based and the numerical-based techniques. In the following paragraphs, we discuss the most relevant directions and the links to our approach.

Approximate Semantics for CRNs. For CRNs including large populations of species, fluid (mean-field) approximation techniques can be applied [5] and extended to approximate higher-order moments [15]: these deterministic approximations lead to a set of ordinary differential equations (ODEs). An alternative is to approximate the CME as a continuous-state stochastic process. The Linear Noise Approximation (LNA) is a Gaussian process which has been derived as an approximation of the CME [16,44] and describes the time evolution of expectation and variance of the species in terms of ODEs. Recently, an aggregation scheme over ODEs that aims at understanding the dynamics of large CRNs has been proposed in [10]. In contrast to our approach, the deterministic approximations cannot adequately capture the stochasticity of CRNs caused by low population species.

To mitigate this drawback, various *hybrid models* have been proposed. The common idea of these models is as follows: the dynamics of low population species is described by the discrete stochastic process and the dynamics of large population species is approximated by a continuous process. The particular hybrid models differ in the approximation of the large population species. In [27], a pure deterministic semantics for large population species is used. The moment-based description for medium/high-copy number species was used in [24]. The LNA approximation and an adaptive partitioning of the species according to leap conditions (that is more general than partitioning based on population thresholds) was proposed in [9]. All hybrid models have to deal with interactions between low and large population species. In particular, the dynamics of the stochastic process describing the low-population species is conditioned by the continuous-state describing the concentration of the large-population species. The numerical analysis of such conditioned stochastic process is typically a computationally demanding task that limits the scalability.

In contrast, our approach does not explicitly partition the species, but rather abstracts the concrete species population using an interval abstraction and tries to effectively capture both the stochastic and the deterministic behaviour with the help of the accelerated transitions. As we already emphasised, the proposed abstraction and analysis avoids any numerical computation of precise quantities.

Reduction Techniques for Stochastic Models. A widely studied reduction method for Markov models is state aggregation based on lumping [6] or (bi-)simulation equivalence [4], with the latter notion in its exact [33] or approximate [13] form. Approximate notions of equivalence have led to new abstraction/refinement techniques for the numerical verification of Markov models over finite [14] as well as uncountably-infinite state spaces [1,41,42]. Several approximate aggregation schemes leveraging the structural properties of CRNs were proposed [17,34,45]. Abate et al. proposed an adaptive aggregation that gives formal guarantees on the approximation error, but typically provide lower state space reductions [2]. Our approach shares the idea of abstracting the state space by aggregating some states together. Similarly to [17,34,45], we partition the state space based on the species population, i.e. we also introduce the population levels. In contrast to the aforementioned aggregation schemes, we propose a

novel abstraction of the transition relation based on the acceleration. It allows us to avoid the numerical solution of the approximate CME and thus achieve a better reduction while providing an accurate predication of the system behaviour.

Alternative methods to deal with large/infinite state spaces are based on a state truncation trying to eliminate insignificant states, i.e., states reached only with a negligible probability. These methods, including finite state projections [36], sliding window abstractions [26], or fast adaptive uniformisation [35], are able to quantify the total probability mass that is lost due to the truncation, but typically cannot effectively handle systems involving a stiff behaviour and multimodality [9].

Simulation-Based Analysis. Transient analysis of CRNs can be performed using the Stochastic Simulation Algorithm (SSA) [21]. Note that the SSA produces a single realisation of the stochastic process, whereas the stochastic solution of CME gives the probability distribution of each species over time. Although simulation-based analysis is generally faster than direct solution of the stochastic process underlying the given CRN, obtaining good accuracy necessitates potentially large numbers of simulations and can be very time consuming.

Various partitioning schemes for species and reactions have been proposed for the purpose of speeding up the SSA in multi-scale systems [23,38,39]. For instance, Yao et al. introduced the slow-scale SSA [7], where they distinguish between fast and slow species. Fast species are then treated assuming they reach equilibrium much faster than the slow ones. Adaptive partitioning of the species has been considered in [19,28]. In contrast to the simulation-based analysis, our approach (i) provides a compact explanation of the system behaviour in the form of tiny models allowing for a synoptic observation and (ii) can easily reveal less probable behaviours.

2   Chemical Reaction Networks

In this paper, we assume familiarity with standard verification of (continuous-time) probabilistic systems, e.g. [4]. For more detail, see [11, Appendix].

CRN Syntax. A *chemical reaction network (CRN)* $\mathcal{N} = (\Lambda, \mathcal{R})$ is a pair of finite sets, where Λ is a set of *species*, $|\Lambda|$ denotes its size, and $\mathcal{R}$ is a set of reactions. Species in Λ interact according to the reactions in $\mathcal{R}$. A *reaction* $\tau \in \mathcal{R}$ is a triple $\tau = (r_\tau, p_\tau, k_\tau)$, where $r_\tau \in \mathbb{N}^{|\Lambda|}$ is the *reactant complex*, $p_\tau \in \mathbb{N}^{|\Lambda|}$ is the *product complex* and $k_\tau \in \mathbb{R}_{>0}$ is the coefficient associated with the rate of the reaction. r_τ and p_τ represent the stoichiometry of reactants and products. Given a reaction $\tau_1 = ([1, 1, 0], [0, 0, 2], k_1)$, we often refer to it as $\tau_1 : \lambda_1 + \lambda_2 \xrightarrow{k_1} 2\lambda_3$.

CRN Semantics. Under the usual assumption of mass action kinetics, the *stochastic* semantics of a CRN $\mathcal{N}$ is generally given in terms of a discrete-state, continuous-time stochastic process $\mathbf{X}(\mathbf{t}) = (X_1(t), X_2(t), \ldots, X_{|\Lambda|}(t), t \geq 0)$ [16]. The *state change* associated to the reaction τ is defined by $\upsilon_\tau = p_\tau - r_\tau$, i.e. the state $\mathbf{X}$ is changed to $\mathbf{X}' = \mathbf{X} + \upsilon_\tau$, which we denote as $\mathbf{X} \xrightarrow{\tau} \mathbf{X}'$. For example,

for τ_1 as above, we have $\upsilon_{\tau_1} = [-1, -1, 2]$. For a reaction to happen in a state $\mathbf{X}$, all reactants have to be in sufficient numbers. The *reachable state space* of $\mathbf{X}(\mathbf{t})$, denoted as $\mathbf{S}$, is the set of all states reachable by a sequence of reactions from a given *initial state* $\mathbf{X}_0$. The set of reactions changing the state $\mathbf{X}_i$ to the state $\mathbf{X}_j$ is denoted as $\mathsf{reac}(\mathbf{X}_i, \mathbf{X}_j) = \{\tau \mid \mathbf{X}_i \xrightarrow{\tau} \mathbf{X}_j\}$.

The behaviour of the stochastic system $\mathbf{X}(\mathbf{t})$ can be described by the (possibly infinite) continuous-time Markov chain (CTMC) $\gamma(\mathcal{N}) = (\mathbf{S}, \mathbf{X}_0, \mathbf{R})$ where the transition matrix $\mathbf{R}(i, j)$ gives the probability of a transition from $\mathbf{X}_i$ to $\mathbf{X}_j$. Formally,

$$\mathbf{R}(i, j) = \sum_{\tau \in \mathsf{reac}(\mathbf{X}_i, \mathbf{X}_j)} k_\tau \cdot C_{\tau, i} \quad \text{where} \quad C_{\tau, i} = \prod_{\ell=1}^{N} \binom{\mathbf{X}_{i, \ell}}{r_\ell} \tag{R}$$

corresponds to the population dependent term of the *propensity function* where $\mathbf{X}_{i, \ell}$ is ℓth component of the state $\mathbf{X}_i$ and r_ℓ is the stoichiometric coefficient of the ℓ-th reactant in the reaction τ. The CTMC $\gamma(\mathcal{N})$ is the accurate representation of CRN $\mathcal{N}$, but—even when finite—not scalable in practice because of the state space explosion problem [25, 31].

3   Semi-quantitative Abstraction

In this section, we describe our abstraction. We derive the desired CTMC conceptually in several steps, which we describe explicitly, although we implement the construction of the final system directly from the initial CRN.

3.1   Over-Approximation by Interval Abstraction and Acceleration

Given a CRN $\mathcal{N} = (\Lambda, \mathcal{R})$, we first consider an interval continuous-time Markov decision process (interval CTMDP[1]), which is a finite abstraction of the infinite $\gamma(\mathcal{N})$. Intuitively, abstract states are given by intervals on sizes of populations with an additional specific that the abstraction captures enabledness of reactions. The transition structure follows the ideas of the standard may abstraction and of the three-valued abstraction of continuous-time systems [30]. A technical difference in the latter point is that we abstract rates into intervals instead of uniformising the chain and then only abstracting transition probabilities into intervals; this is necessary in later stages of the process. The main difference is that we also treat certain sequences of actions, which we call acceleration.

Abstract Domains. The first step is to define the abstract domain for the population sizes. For every species $\lambda \in \Lambda$, we define a finite partitioning A_λ of $\mathbb{N}$ into intervals, reflecting the rough size of the population. Moreover, we want the abstraction to reflect whether a reaction is enabled. Hence we require that

---

[1] Interval CTMDP is a CTMDP with lower/upper bounds on rates. Since it serves only as an intermediate formalism to ease the presentation, we refrain from formalising it here.

$\{0\} \in A_\lambda$ for the case when the coefficients of this species as a reactant is always 0 or 1; in general, for every $i < \max_{\tau \in \mathcal{R}} r_\tau(\lambda)$ we require $\{i\} \in A_\lambda$.

The abstraction $\alpha_\lambda(n)$ of a number n of a species λ is then the $I \in A_\lambda$ for which $n \in I$. The state space of $\alpha(\mathcal{N})$ is the product $\prod_{\lambda \in \Lambda} A_\lambda$ of the abstract domains with the point-wise defined abstraction $\alpha(\boldsymbol{n})_\lambda = \alpha_\lambda(\boldsymbol{n}_\lambda)$.

The abstract domain for the rates according to (R) is the set of all real intervals.

Transitions from an abstract state are defined as the may abstraction as follows. Since our abstraction reflect enabledness, the same set of action is enabled in all concrete states of a given abstract state. The targets of the action in the abstract setting are abstractions of all possible concrete successors, i.e. $succ(s, a) := \{\alpha(\boldsymbol{n}) \mid \boldsymbol{m} \in s, \boldsymbol{m} \xrightarrow{a} \boldsymbol{n}\}$, in other words, the transitions enabled in at least one of the respective concrete states. The abstract rate is the smallest interval including all the concrete rates of the respective concrete transitions. This can be easily computed by the corner-points abstraction (evaluating only the extremum values for each species) since the stoichiometry of the rates is monotone in the population sizes.

High-Level of Non-determinism. The (more or less) standard style of the abstraction above has several drawbacks—mostly related to the high degree of non-determinism for rates—which we will subsequently discuss.

Firstly, in connection with the abstract population sizes, transitions to different sizes only happen non-deterministically, leaving us unable to determine which behaviour is probable. For example, consider the simple system given by $\lambda \xrightarrow{d} \emptyset$ with $k_d = 10^{-4}$ so the degradation happens on average each 10^4 seconds. Assume population discretisation into $[0], [1..5], [6..20], [21..\infty)$ with abstraction depicted in Fig. 1. While the original system obviously moves from $[6..20]$ to $[1..5]$ very probably in less than $15 \cdot 10^4$ seconds, the abstraction cannot even say that it happens, not to speak of estimating the time.
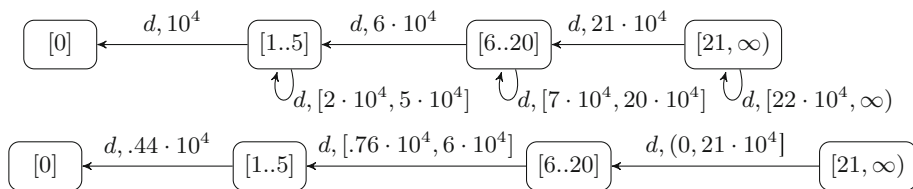

Fig. 1. Above: Interval CTMDP abstraction with intervals on rates and non-determinism. Below: Interval CTMC abstraction arising from acceleration.

Acceleration. To address this issue, we drop the non-deterministic self-loops and transitions to higher/lower populations in the abstract system.[2] Instead,

---

[2] One can also preserve the non-determinism for the special case when one of the transitions leads to a state where some action ceases to be enabled. While this adds more precision, the non-determinism in the abstraction makes it less convenient to handle.

we *"accelerate"* their effect: We consider sequences of these actions that in the concrete system have the effect of changing the population level. In our example above, we need to take the transition 1 to 13 times from [6..20] with various rates depending on the current concrete population, in order to get to [1..5]. This makes the precise timing more complicated to compute. Nevertheless, the expected time can be approximated easily: here it ranges from $\frac{1}{6} \cdot 10^4 = 0.17 \cdot 10^4$ (for population 6) to roughly $(\frac{1}{20} + \frac{1}{19} + \cdots + \frac{1}{6}) \cdot 10^4 = 1.3 \cdot 10^4$ (for population 20). This results in an interval CTMC.[3]

Concurrency in Acceleration. The accelerated transitions can due to higher number of occurrences be considered continuous or deterministic, as opposed to discrete stochastic changes as distinguished in the hybrid approach. The usual differential equation approach would also take into account other reactions that are modelled deterministically and would combine their effect into one equation. In order to simplify the exposition and computation and—as we see later— without much loss of precision, we can consider only the fastest change (or non-deterministically more of them if their rates are similar).[4]

3.2   Operational Semantics: Concretisation to a Representative

The next disadvantage of classical abstraction philosophy, manifested in the interval CTMC above is that the precise-valued intervals on rates imply high computational effort during the analysis. Although the system is smaller, standard transient analysis is still quite expensive.

Concretisation. In order to deal with this issue, the interval can be approximated roughly by the expected time it would take for an average population in the considered range, in our example the "average" representative is 13. Then the first transition occurs with rate $13 \cdot 10^{-4} = 10^{-3}$ and needs to happen 7 times, yielding expected time $7/13 \cdot 10^4 = 0.5 \cdot 10^4$ (ignoring even the precise slow downs in the rates as the population decreases). Already this very rough computation yields relative precision with factor 3 for all the populations in this interval, thus yielding the correct order of magnitude with virtually no effort. We lift the concretisation naturally to states and denote the concretisation of abstract state s by $\gamma(s)$. The complete procedure is depicted in Algorithm 1.

The concretisation is one of the main points where we deliberately drop a lot of quantitative information, while still preserving some to conclude on big quantitative differences. Of course, the precision improves with more precise abstract domains and also with higher differences on the original rates.

---

[3] The waiting times are not distributed according to the rates in the intervals. It is only the expected waiting time (reciprocal of the rate) that is preserved. Nevertheless, for ease of exposition, instead of labelling the transitions with expected waiting times we stick to the CTMC style with the reciprocals and formally treat it as if the label was a real rate.

[4] Typically the classical concurrency diamond appears and the effect of the other accelerated reactions happen just after the first one.

Algorithm 1. Semi-quantitative abstraction CTMC $\alpha(\mathcal{N})$

1: $A \leftarrow \prod_{\lambda \in \Lambda} A_\lambda$	▷ States
2: **for** $\boldsymbol{a} \in A$ **do**	▷ Transitions
3: $\boldsymbol{c} \leftarrow \gamma(\boldsymbol{a})$	▷ Concrete representative
4: **for** each τ enabled in $\boldsymbol{c}$ **do**	
5: $r \leftarrow$ rate of τ in $\boldsymbol{c}$	▷ According to (R)
6: $\boldsymbol{a}' \leftarrow \alpha(\boldsymbol{c} + v_\tau)$	▷ Successor
7: set $\boldsymbol{a} \xrightarrow{\tau} \boldsymbol{a}'$ with rate r	
8: **for** self-loop $\boldsymbol{a} \xrightarrow{\tau} \boldsymbol{a}$ **do**	▷ Accelerate self-loops
9:          $n_\tau \leftarrow \min\{n \mid \alpha(\boldsymbol{c} + n \cdot v_\tau) \neq \boldsymbol{a}\}$ ▷ the number of τ to change the abstract state
10: $\boldsymbol{a}' \leftarrow \alpha(\boldsymbol{c} + n_\tau \cdot v_\tau)$	▷ Acceleration successor
11:         instead of the self-loop with rate r, set $\boldsymbol{a} \xrightarrow{\tau} \boldsymbol{a}'$ with rate $n_\tau \cdot r$

It remains to determine the representative for the unbounded interval. In order to avoid infinity, we require an additional input for the analysis, which are deemed upper bounds on possible population of each species. In cases when any upper bound is hard to assume, we can analyse the system with a random one and see if the last interval is reachable with significant probability. If yes, then we need to use this upper bound as a new point in the interval partitioning and try a higher upper bound next time. In general, such conditions can be checked in the abstraction and their violation implies a recommendation to refine the abstract domains accordingly.

Orders-of-Magnitude Abstraction. Such an approximation is thus sufficient to determine most of the time whether the acceleration (sequence of actions) happens sooner or later than e.g. another reaction with rate 10^{-6} or 10^{-2}. Note that this *decision* gets more precise not only as we refine the population levels, but also as the system gets stiffer (the concrete values of the rates differ more), which are normally harder to analyse. For the ease of presentation in our case studies, we shall depict only the magnitude of the rates, i.e. the decadic logarithm rounded to an integer.

Non-determinism and Refinement. If two rates are close to each other, say of the same magnitude (or difference 1), such a rough computation (and rough population discretisation) is not precise enough to determine which of the reactions happens with high probability sooner. Both may be happening roughly at the same pace, or with more information we could conclude one of them is considerably faster. This introduces an uncertainty, showing different behaviours are possible depending on the exact quantities. This indicates points where refinement might be needed if more precise results are required. For instance, with rates of magnitudes 2 and 3, the latter should be happing most of the time, the former only with a few percent chance. If we want to know whether it is rather tens of percent or tenths of percent, we should refine the abstraction.

4   Semi-quantitative Analysis

In this section, we present an approximative analysis technique that describes the most probable transient and steady-state behaviour of the system (also with rough timing) and on demand also the (one or more orders of magnitude) less probable behaviours. As such it is robust in the sense that it is well suited to work with imprecise rates and populations. It is computationally easy (can be done in hand in time required for a computer by other methods), while still yielding significant quantitative results ("in orders of magnitude"). It does not provide exact error guarantees since computing them would be almost as expensive as the classical analysis. It only features trivial limit-style bounds: if the population abstraction gets more and more refined, the probabilities converge to those of the original system; further, the higher the separation between the rate magnitudes, the more precise the approximation is since the other factors (and thus the incurred imprecisions) play less significant role.

Intuitively, the main idea—similar to some multi-rate simulation techniques for stiff systems—is to "simulate" "fast" reactions until the steady state and then examine which slower reactions take place. However, "fast" does not mean faster than some constant, but faster than other transitions in a given state. In other words, we are not distinguishing fast and slow reactions, but tailor this to each state separately. Further, "simulation" is not really a stochastic simulation, but a deterministic choice of the fastest available transition. If a transition is significantly faster than others then this yields what a simulation would yield. When there are transitions with similar rates, e.g. with at most one order of magnitude difference, then both are taken into account as described in the following definition.

Pruned System. Consider the underlying graph of the given CTMC. If we keep only the outgoing transitions with the maximum rate in each state, we call the result *pruned*. If there is always (at most) one transition then the graph consists of several paths leading to cycles. In general when more transitions are kept, it has bottom strongly connected components (bottom SCCs, BSCCs) and some transient parts.

We generalise this concept to *n-pruning* that preserves all transitions with a rate that is not more than n orders of magnitude smaller than the maximum rate in the state. Then the pruning above is 0-pruning, 1-pruning preserves also transitions happening up to 10 times slower, which can thus still happen with dozens of percent, 2-pruning is relevant for analysis where behaviour occurring with units of percent is also tracked etc.

Algorithm Idea. Here we explain the idea of Algorithm 2. The transient parts of the pruned system describe the most probable behaviour from each state until the point where visited states start to repeat a lot (steady state of the pruned system). In the original system, the usual behaviour is then to stay in this SCC C until one of the pruned (slower) reactions occurs, say from state s to state t. This may bring us to a different component of the pruned graph and the analysis process repeats. However, t may also bring us back into C, in which case we stay

in the steady-state, which is basically the same as without the transition from s to t. Further, t might be in the transient part leading to C, in which case these states are added to C and the steady state changes a bit, spreading the distribution slightly also to the previously transient states. Finally, t might be leading us into a component D where this run was previous to visiting C. In that case, the steady-state distribution spreads over all the components visited between D and C, putting a probability mass to each with a different order of magnitude depending on all the (magnitudes of) sojourn times in the transient and steady-state phases on the way.

Using the macros defined in the algorithm, the correctness of the computations can be shown as follows. For the time spent in the transient phase (line 16), we consider the slowest sojourn time on the way times the number of such transitions; this is accurate since the other times are by order(s) of magnitude shorter, hence negligible. The steady-state distribution on a BSCC of the

---

Algorithm 2. Semi-quantitative analysis

1: $W \leftarrow \emptyset$ ▷ worklist of SCCs to process
2: add {initial state} to W and assign iteration 0 to it   ▷ artificial SCC to start the process
3: **while** $W \neq \emptyset$ **do**
4:      $C \leftarrow$ pop W
                   ▷ Compute and output steady state or its approximation
5:      steady-state of C is approximately $minStayingRate/(m \cdot stayingRate(\cdot))$
6:      **if** C has no exits **then** continue   ▷ definitely bottom SCC, final steady state
           ▷ Compute and output exiting transitions and the time spent in C
7:      $exitStates \leftarrow \arg\min_C(stayingRate(\cdot)/exitingRate(\cdot))$ ▷ Probable exit points
8:      $minStayingRate \leftarrow$ minimum rate in C, $m \leftarrow$#occurrences there
9:      $timeToExit \leftarrow stayingRate(s) \cdot m/(|exitStates| \cdot minStayingRate \cdot exitingRate(s))$
                      for (arbitrary) $s \in exitStates$
10:      **for all** $s \in exitsStates$ **do**   ▷ Transient analysis
11:         $t \leftarrow$ target of the exiting transition
12:         $T \leftarrow$ SCCs reachable in the pruned graph from t
13:         thereby newly reached transitions get assigned iteration of $C + 1$
14:         **for** $D \in T$ **do**
                    ▷ Compute and output time to get from t to D
15:            $minRate \leftarrow$ minimum rate on the way from t to D, $m \leftarrow$#occurrences there
16:            $transTime \leftarrow m/minRate$
                         ▷ Determine the new SCC
17:           **if** $D = C$ **then**   ▷ back to the current SCC
18:             add to W the union of C and the new transient path τ from t to C
19:             in later steady-state computation, the states of τ will have probability
                  smaller by a factor of $stayingRate(s)/exitingRate(s)$
20:           **else if** D was previously visited **then**   ▷ alternating between different SCCs
21:             add to W the merge of all SCCs visited between D and C (inclusively)
22:             in later steady-state computation, reflect all $timeToExit$ and $transTime$
                       between D and C
23:           **else**   ▷ new SCC
24:             add D to W

MACROS:

$stayingRate(s)$ is the rate of transitions from s in the pruned graph
$exitingRate(s)$ is the maximum rate of transitions from s not in the pruned graph

pruned graph can be approximated by the $minStayingRate/(m \cdot stayingRate(\cdot))$ on line 5. Indeed, it corresponds to the steady-state distribution if the BSCC is a cycle and the $minStayingRate$ significantly larger than other rates in the BSCC since then the return time for the states is approximately $m/minStayingRate$ and the sojourn time $1/stayingRate(\cdot)$. The component is exited from s with the proportion given by its steady-state distribution times the probability to take the exit during that time. The former is approximated above; the latter can be approximated by the density in 0, i.e. by $exitingRate(s)$, since the staying rate is significantly faster. Hence the candidates for exiting are maximising $exitingRate(\cdot)/stayingRate(\cdot)$ as on line 7. There are $|exitStates|$ candidates for exit and the time to exit the component by a particular candidate s is the expected number of visits before exit, i.e. $stayingRate(s) \cdot exitingRate(s)$ times the return time $m \cdot minStayingRate$, hence the expression on line 9.
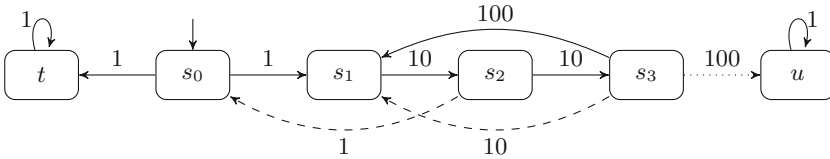

Fig. 2. Alternating transient and steady-state analysis.

For example, consider the system in Fig. 2. Iteration 1 reveals the part with solid lines with two (temporary) BSCCs $\{t\}$ and $\{s_1, s_2, s_3\}$. The former turns out definitely bottom. The latter has a steady state proportional to $(10^{-1}, 10^{-1}, 100^{-1})$. Its most probable exits are the dashed ones, identified in the subsequent iteration 2, probable proportionally to $(1/10, 10/100)$; the expected time to take them is $10 \cdot 2/(2 \cdot 10 \cdot 1) = 1 = 100 \cdot 2/(2 \cdot 10 \cdot 10)$. The latter leads back to the current SCC and does not change the set of BSCCs (hence in our examples below we often either skip or merge such iterations for the sake of readability). In contrast, the former leads to a previous SCC; thereafter $\{s_1, s_2, s_3\}$ is no more a bottom SCC and consequently the third exit to u is not even analysed. Nevertheless, it could still happen with minor probability, which can be seen if we consider 1-pruning instead.

5   Experimental Evaluation and Discussion

In order to demonstrate the applicability and accuracy of our approach, we selected the following three biologically relevant case studies. (1) stochastic model of gene expression [22,24], (2) Goutsias's model [23] describing transcription regulation of a repressor protein in bacteriophage λ and (3) viral infection model [43].

Although the underlying CRNs are quite small (up to 5 species and 10 reaction), their analysis is very challenging: (i) the stochasticity has a strong impact

on the dynamics of these systems and thus purely deterministic approximations via ODEs are not accurate, (ii) the systems include species with low, medium, and high populations and thus the resulting state space of the stochastic process is prohibitively large to perform precise numerical analysis and existing reduction/approximation techniques are not sufficient (they are either too imprecise or do not provide sufficient reduction factors), and (iii) the system dynamics leads to bi-modal distributions and/or is affected by stiff reactions.

These models thus represent perfect candidates for evaluating advanced approximation methods including various hybrid approaches [9,24,27]. Although these approaches can handle the models, they typically require tens of minutes or hours of computation time. Similarly simulation-based methods are very time consuming especially in case of very stiff CRN, represented by the viral infection model. We demonstrate that our approach provides accurate predications of the system behaviour and is feasible even when performed manually by a human.

Recall that the algorithm that builds the abstract model of the given CRN takes as input two vectors representing the population discretisation and population bounds. We generally assume that these inputs are provided by users who have a priori knowledge about the system (e.g. in which orders the species population occurs) and that the inputs also reflect the level of details the users are interested in. In the following case studies, we, however, set the inputs only based on the rate orders of the reactions affecting the particular species (unless mentioned otherwise).

5.1 Gene Expression Model

The CRN underlying the gene expression model is described in Table 1. As discussed in [24] and experimentally observed in [18], the system oscillates between two phases characterised by the D_{on} state and the D_{off} state, respectively. Biologists are interested in how the distribution of the D_{on} and D_{off} states is aligned with the distribution of RNA and proteins P, and how the correlation among the distributions depends on the DNA switching rates.

The state vector of the underlying CTMC is given as [P, RNA, D_{off}, D_{on}]. We use very relaxed bounds on the maximal populations, namely the bound 1000 for P and 100 for RNA. Note the DNA invariant $D_{on} + D_{off} = 1$. As in [24], the initial state is given as [10,4,1,0].

We first consider the slow switching rates that lead to a more complicated dynamics including bimodal distributions. In order to demonstrate the refinement step and its effect on the accuracy of the model, we start with a very coarse abstraction. It distinguishes only the zero population and the non-zero populations and thus it is not able to adequately capture the relationship between the DNA state and RNA/P population. The pruned abstract model obtained using Algorithm 1 and 2 is depicted in Fig. 3 (left). The full one before pruning is shown in Fig. 6 [11, Appendix].

The proposed analysis of the model identifies the key trends in the system dynamic. The red transitions, representing iterations 1–3, capture the most probable paths in the system. The green component includes states with DNA on

Table 1. Gene expression. For slow DNA switching, $r_1 = r_2 = 0.05$. For fast DNA switching, $r_1 = r_2 = 1$. The rates are in h^{-1}.
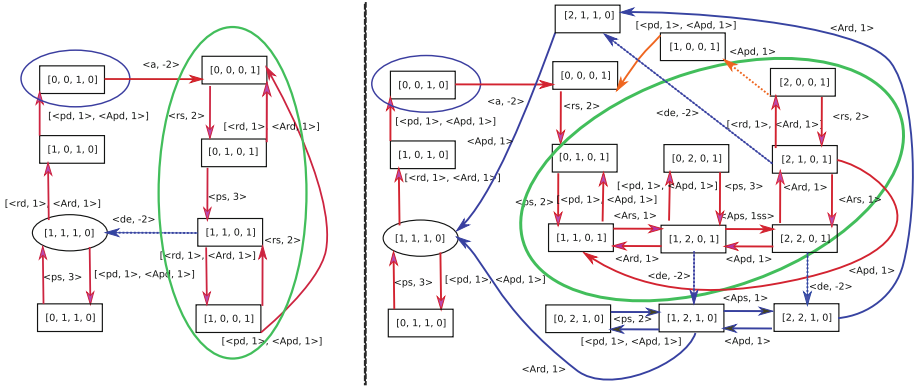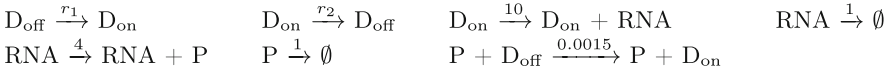
$$D_{off} \xrightarrow{r_1} D_{on} \qquad D_{on} \xrightarrow{r_2} D_{off} \qquad D_{on} \xrightarrow{10} D_{on} + RNA \qquad RNA \xrightarrow{1} \emptyset$$

$$RNA \xrightarrow{4} RNA + P \qquad P \xrightarrow{1} \emptyset \qquad P + D_{off} \xrightarrow{0.0015} P + D_{on}$$

Fig. 3. Pruned abstraction for the gene expression model using the coarse population discretisation (left) and after the refinement (right). The state vector is [P, RNA, D_{off}, D_{on}].

(i.e. $D_{on} = 1$) where the system oscillates. The component is reached via the blue state with D_{off} and no RNAs/P. The blue state is promptly reached from the initial state and then the system waits (roughly 100 h according our rate abstraction) for the next DNA activation. The oscillation is left via a deactivation in the iteration 4 (the blue dotted transition)[5]. The estimation of the exit time computed using Algorithm 2 is also 100 h. The deactivation is then followed by fast red transitions leading to the blue state, where the system waits for the next activation. Therefore, we obtain an oscillation between the blue state and the green component, representing the expected oscillation between the D_{on} and D_{off} states.

As expected, this abstraction does not clearly predict the bimodal distribution on the RNA/P populations as the trivial population levels do not bear any information beside reaction enabledness. In order to obtain a more accurate analysis of the system, we refine the population discretisation using a single level threshold for P and DNA, that is equal to 100 and 10, respectively (the rates in the CRN indicate that the population of P reaches higher values).

Figure 3 (right) depicts the pruned abstract model with the new discretisation (the full model is depicted in Fig. 7 [11, Appendix]. We again obtain the oscillation between the green component representing DNA$_{on}$ states and the blue DNA$_{off}$ state. The states in the green component more accurately predicts

---

[5] In Fig. 3, the dotted transitions denote exit transitions representing the deactivations.

that in the DNA_{on} states the populations of RNA and P are high and drop to zero only for short time periods. The figure also shows orange transitions within the iteration 2 that extend the green component by two states. Note that the system promptly returns from these states back to the green component. After the deactivation in the iteration 4, the system takes (within the same iteration) the fast transitions (solid blue) leading to the blue component where system waits for another activation and where the mRNA/protein populations decrease. The expected time spent in states on blue solid transitions is small and thus we can reliably predict the bimodal distribution of the mRNA/P populations and its correlation with the DNA state. The refined abstraction also reveals that the switching time from the DNA_{on} mode to the DNA_{off} mode is lower. These predications are in accordance with the results obtained in [24]. See Fig. 8 [11, Appendix] that is adopted from [24] and illustrates these results.

To further test the accuracy of our approach, we consider the fast switching between the DNA states. We follow the study in [24] and increase the rates by two orders of magnitude. We use the refined population discretisation and obtain a very similar abstraction as in Fig. 3 (right). We again obtain the oscillation between the green component (DNA_{on} states and nonzero RNA/protein populations) and the blue state (DNA_{off} and zero RNA/protein populations). The only difference is in fact the transition rates corresponding to the activation and deactivation causing that the switching rate between the components is much faster. As a consequence, the system spends a longer period in the blue transient states with D_{off} and nonzero RNA/protein populations. The time spent in these states decreases the correlation between the DNA state and the RNA/protein populations as well as the bimodality in the population distribution. This is again in the accordance with [24].

To conclude this case study, we observe a very aligned agreement between the results obtained using our approach and results in [24] obtained via advanced and time consuming numerical methods. We would like to emphasise that our abstraction and its solution is obtained within a fraction of a second while the numerical methods have to approximate solutions of equations describing high-order conditional moments of the population distributions. As [24] does not report the runtime of the analysis and the implementation of their methods is not publicly available, we cannot directly compare the time complexity.

5.2   Goutsias's Model

Goutsias's model illustrated in Table 2 is widely used for evaluation of various numerical and simulation based techniques. As showed e.g. in [23], the system has with a high probability the following transient behaviour. In the first phase, the system switches with a high rate between the non-active DNA (denoted DNA) and the active DNA (DNA.D). During this phase the population of RNA, monomers (M) and dimers (D) gradually increase (with only negligible oscillations). After around 15 min, the DNA is blocked (DNA.2D) and the population of RNA decreases while the population of M and D is relatively stable. After all RNA degrades (around another 15 min) the system switches to the third
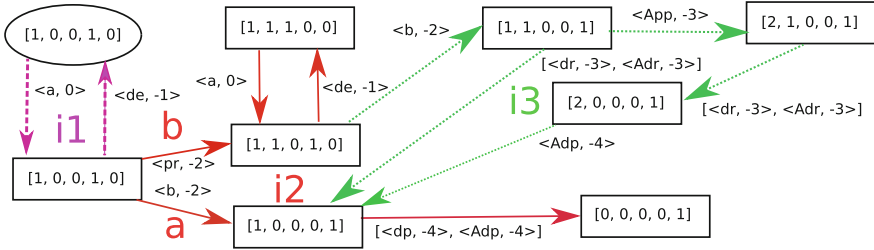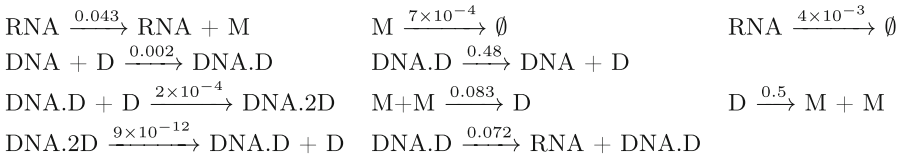
Table 2. Goutsias' Model. The rates are in s^{-1}

$$\text{RNA} \xrightarrow{0.043} \text{RNA} + \text{M} \qquad\qquad \text{M} \xrightarrow{7\times10^{-4}} \emptyset \qquad\qquad\qquad \text{RNA} \xrightarrow{4\times10^{-3}} \emptyset$$

$$\text{DNA} + \text{D} \xrightarrow{0.002} \text{DNA.D} \qquad \text{DNA.D} \xrightarrow{0.48} \text{DNA} + \text{D}$$

$$\text{DNA.D} + \text{D} \xrightarrow{2\times10^{-4}} \text{DNA.2D} \qquad \text{M+M} \xrightarrow{0.083} \text{D} \qquad\qquad \text{D} \xrightarrow{0.5} \text{M} + \text{M}$$

$$\text{DNA.2D} \xrightarrow{9\times10^{-12}} \text{DNA.D} + \text{D} \qquad \text{DNA.D} \xrightarrow{0.072} \text{RNA} + \text{DNA.D}$$

Fig. 4. Pruned abstraction for the Goutsias' model. The state vector is [M + D, RNA, DNA, DNA.D, DNA.2D]

phase where the population of M and D slowly decreases. Further, there is a non-negligible probability that the DNA is blocked at the beginning while the population of RNA is still small and the system promptly dies out.

Although the system is quite suitable for the hybrid approaches (there is no strong bimodality and only a limited stiffness), the analysis still takes 10 to 50 min depending on the required precision [27]. We demonstrate that our approach is able to accurately predict the main transient behaviour as well as the non-negligible probability that the system promptly dies out.

The state vector is given as [M, D, RNA, DNA, DNA.D, DNA.2D] and the initial state is set to [2, 6, 0, 1, 0, 0] as in [27]. We start our analysis with a coarse population discretisation with a single threshold 100 for M and D and a single threshold 10 for RNA. We relax the bounds, in particular, 1000 for M and D, and 100 for RNA. Note that these numbers were selected solely based on the rate orders of the relevant reactions. Note the DNA invariant DNA + DNA.D + DNA.2D = 1.

Figure 4 illustrates the pruned abstract model we obtained (the full model is depicted in Fig. 9 [11, Appendix]. For a better visualisation, we merged the state components corresponding to M and D into one component with M + D. As there is the fast reversible dimerisation, the actual distributions between the population of M and D does not affect the transient behaviour we are interested in.

The analysis of the model shows the following transient behaviour. The purple dotted loop in the iteration i1 represents (de-)activation of the DNA. The expected exit time of this loop is 100 s. According to our abstraction, there are two options (with the same probability) to exit the loop: (1) the path a rep-

resents the DNA blocking followed by the quick extinction and (2) the path b corresponds to the production of RNA and its followed by the red loop in the i2 that again represents (de-)activation of the DNA. Note that according our abstraction, this loop contains states with the populations of M/D as well as RNA up to 100 and 10, respectively.

The expected exit time of this loop is again 100 s and there are two options how to leave the loop: (1) the path within the iteration $i3$ (taken with roughly 90%) represents again the DNA blocking and it is followed by the extension of RNA and consequently by the extension of M/D in about 1000 s and (2) the path within the iteration 5 (shown in the full graph in Fig. 9 [11, Appendix]) taken with roughly 10% represents the series of protein productions and leads to the states with a high number of proteins (above 100 in our population discretisation). Afterwards, there is again a series of DNA (de-)activations followed by the DNA blocking and the extinction of RNA. As before, this leads to the extinction of M/D in about 1000 s.

Although this abstraction already shows the transient behaviour leading to the extinction in about 30 min, it introduces the following inaccuracy with respect to the known behaviour: (1) the probability of the fast extinction is higher and (2) we do not observe the clear bell-shape pattern on the RNA (i.e. the level 2 for the RNA is not reached in the abstraction). As in the previous case study, the problem is that the population discretisation is too coarse. It causes that the total rate of the DNA blocking (affected by the M/D population via the mass action kinetics) is too high in the states with the M/D population level 1. This can be directly seen in the interval CTMC representation where the rate spans many orders of magnitude, incurring too much imprecision. The refinement of the M/D population discretisation eliminates the first inaccuracy. To obtain the clear bell-shape patter on RNA, one has to refine also the RNA population discretisation.

5.3   Viral Infection

The viral infection model described in Table 3 represents the most challenging system we consider. It is highly stochastic, extremely stiff, with all species presenting high variance and some also very high molecular populations. Moreover, there is a bimodal distribution on the RNA population. As a consequence, the solution of the full CME, even using advanced reduction and aggregation techniques, is prohibitive due to state-space explosion and stochastic simulation are very time consuming. State-of-the-art hybrid approaches integrating the LNA and an adaptive population partitioning [9] can handle this system but also need a very long execution time. For example, a transient analysis up to time $t = 50$ requires around 20 min and up to $t = 200$ more than an hour.

To evaluate the accuracy of our approach on this challenging model, we also focus on the same transient analysis, namely, we are interested in the distribution of RNA at time $t = 200$. The analysis in [9] predicts a bimodal distribution where, the probability that RNA is zero in around 20% and the remaining probability has Gaussian distribution with mean around 17 and the probability that there

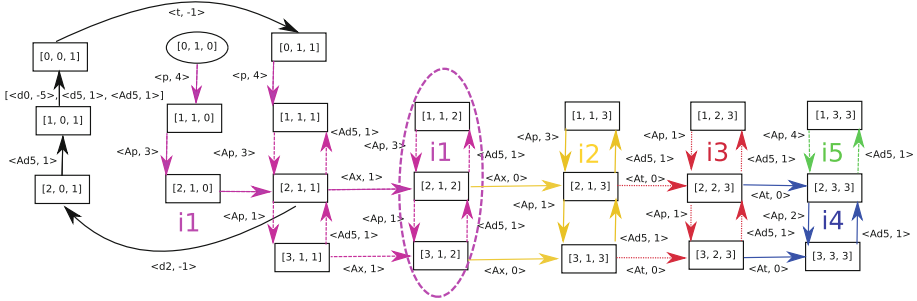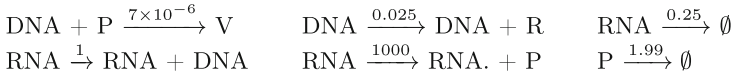**Table 3.** Viral Infection. The rates are day^{-1}

$$\text{DNA} + \text{P} \xrightarrow{7 \times 10^{-6}} \text{V} \qquad \text{DNA} \xrightarrow{0.025} \text{DNA} + \text{R} \qquad \text{RNA} \xrightarrow{0.25} \emptyset$$

$$\text{RNA} \xrightarrow{1} \text{RNA} + \text{DNA} \qquad \text{RNA} \xrightarrow{1000} \text{RNA.} + \text{P} \qquad \text{P} \xrightarrow{1.99} \emptyset$$

Fig. 5. Pruned abstraction for the viral infection model. The state vector is [P, RNA, DNA].

is more than 30 RNAs is close to zero. This is confirmed by simulation-based analysis in [23] showing also the gradual growth of the RNA population. The simulation-based analysis in [43], however, estimates a lower probability (around 3%) that RNA is 0 and higher mean of the remaining Gaussian distribution (around 23). Recall that obtaining accurate results using simulations is extremely time consuming due to very stiff reactions (a single simulation for $t = 200$ takes around 20 s).

In the final experiments, we analyse the distribution of RNA at time $t = 200$ using our approach. The state vector is given as [P, RNA, DNA] and we start with the concrete state [0, 1, 0]. To sufficiently reason about the RNA population and to handle the very high population of the proteins, we use the following population discretisation: thresholds {10, 1000} for P, {10, 30} for RNA, and {10, 100} for DNA. As before, we use very relaxed bounds 10000, 100, and 1000 for P, RNA, and D, respectively. Note that we ignore the population of the virus V as it does not affect the dynamics of the other species. This simplification makes the visualisation of our approach more readable and has no effect on the complexity of the analysis.

Figure 5 illustrates the obtained abstract model enabling the following transient analysis (the full model is depicted in Fig. 10 [11, Appendix]. In a few days the system reaches from the initial state the loop (depicted by the purple dashed ellipse) within the iteration *i1*. The loop includes states where RNA has level 1, DNA has level 2 and P oscillates between the levels 2 and 3. Before entering the loop, there is a non-negligible probability (orders of percent) that the RNA drops to 0 via the full black branch that returns to transient part of the loop in *i1*. In this branch the system can also die out (not shown in this figure, see the full model) with probability in the order of tenths of percent.

The average exit time of the loop in *i1* is in the order of 10 days and the system goes to the yellow loop within the iteration *i2*, where the DNA level is increased to 3 (RNA level is unchanged and P again oscillates between the levels 2 and 3). The average exit time of the loop in *i2* is again in the order of 10 days and systems goes to the dotted red loop within iteration *i3*. The transition represents the sequence of RNA synthesis that leads to RNA level 2. P oscillates as before. Finally, the system leaves the loop in *i3* (this takes another dozen days) and reaches RNA level 3 in iterations *i4* and *i5* where the DNA level remains at the level 3 and P oscillates. The iteration *i4* and *i5* thus roughly correspond to the examined transient time $t = 200$.

The analysis clearly demonstrates that our approach leads to the behaviour that is well aligned with the previous experiments. We observed growth of the RNA population with a non-negligible probability of its extinction. The concrete quantities (i.e. the probability of the extinction and the mean RNA population) are closer to the analysis in [43]. The quantities are indeed affected by the population discretisation and can be further refined. We would like to emphasise that in contrast to the methods presented in [9,23,43] requiring hours of intensive numerical computation, our approach can be done even manually on the paper.

References

1. Abate, A., Katoen, J.P., Lygeros, J., Prandini, M.: Approximate model checking of stochastic hybrid systems. Eur. J. Control **16**, 624–641 (2010)
2. Abate, A., Brim, L., Češka, M., Kwiatkowska, M.: Adaptive aggregation of Markov chains: quantitative analysis of chemical reaction networks. In: Kroening, D., Păsăreanu, C.S. (eds.) CAV 2015. LNCS, vol. 9206, pp. 195–213. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21690-4_12
3. Angluin, D., Aspnes, J., Eisenstat, D., Ruppert, E.: The computational power of population protocols. Distrib. Comput. **20**(4), 279–304 (2007)
4. Baier, C., Katoen, J.P.: Principles of Model Checking. The MIT Press, Cambridge (2008)
5. Bortolussi, L., Hillston, J.: Fluid model checking. In: Koutny, M., Ulidowski, I. (eds.) CONCUR 2012. LNCS, vol. 7454, pp. 333–347. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32940-1_24
6. Buchholz, P.: Exact performance equivalence: an equivalence relation for stochastic automata. Theor. Comput. Sci. **215**(1–2), 263–287 (1999)
7. Cao, Y., Gillespie, D.T., Petzold, L.R.: The slow-scale stochastic simulation algorithm. J. Chem. Phys. **122**(1), 014116 (2005)
8. Cardelli, L.: Two-domain DNA strand displacement. Math. Struct. Comput. Sci. **23**(02), 247–271 (2013)
9. Cardelli, L., Kwiatkowska, M., Laurenti, L.: A stochastic hybrid approximation for chemical kinetics based on the linear noise approximation. In: Bartocci, E., Lio, P., Paoletti, N. (eds.) CMSB 2016. LNCS, vol. 9859, pp. 147–167. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45177-0_10
10. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Maximal aggregation of polynomial dynamical systems. Proc. Natl. Acad. Sci. **114**(38), 10029–10034 (2017)

11. Češka, M., Křetínský, J.: Semi-quantitative abstraction and analysis of chemical reaction networks. Technical report abs/1905.09914, arXiv.org (2019)
12. Chellaboina, V., Bhat, S.P., Haddad, W.M., Bernstein, D.S.: Modeling and analysis of mass-action kinetics. IEEE Control Syst. Mag. **29**(4), 60–78 (2009)
13. Desharnais, J., Laviolette, F., Tracol, M.: Approximate analysis of probabilistic processes: logic, simulation and games. In: Quantitative Evaluation of SysTems (QEST), pp. 264–273. IEEE (2008)
14. D'Innocenzo, A., Abate, A., Katoen, J.P.: Robust PCTL model checking. In: Hybrid Systems: Computation and Control (HSCC), pp. 275–285. ACM (2012)
15. Engblom, S.: Computing the moments of high dimensional solutions of the master equation. Appl. Math. Comput. **180**(2), 498–515 (2006)
16. Ethier, S.N., Kurtz, T.G.: Markov Processes: Characterization and Convergence, vol. 282. Wiley, New York (2009)
17. Ferm, L., Lötstedt, P.: Adaptive solution of the master equation in low dimensions. Appl. Numer. Math. **59**(1), 187–204 (2009)
18. Gandhi, S.J., Zenklusen, D., Lionnet, T., Singer, R.H.: Transcription of functionally related constitutive genes is not coordinated. Nat. Struct. Mol. Biol. **18**(1), 27 (2011)
19. Ganguly, A., Altintan, D., Koeppl, H.: Jump-diffusion approximation of stochastic reaction dynamics: error bounds and algorithms. Multiscale Model. Simul. **13**(4), 1390–1419 (2015)
20. Giacobbe, M., Guet, C.C., Gupta, A., Henzinger, T.A., Paixão, T., Petrov, T.: Model checking gene regulatory networks. In: Baier, C., Tinelli, C. (eds.) TACAS 2015. LNCS, vol. 9035, pp. 469–483. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46681-0_47
21. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. **81**(25), 2340–2361 (1977)
22. Golding, I., Paulsson, J., Zawilski, S.M., Cox, E.C.: Real-time kinetics of gene activity in individual bacteria. Cell **123**(6), 1025–1036 (2005)
23. Goutsias, J.: Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems. J. Chem. Phys. **122**(18), 184102 (2005)
24. Hasenauer, J., Wolf, V., Kazeroonian, A., Theis, F.: Method of conditional moments (MCM) for the chemical master equation. J. Math. Biol. 1–49 (2013). https://doi.org/10.1007/s00285-013-0711-5
25. Heath, J., Kwiatkowska, M., Norman, G., Parker, D., Tymchyshyn, O.: Probabilistic model checking of complex biological pathways. Theor. Comput. Sci. **391**(3), 239–257 (2008)
26. Henzinger, T.A., Mateescu, M., Wolf, V.: Sliding window abstraction for infinite Markov chains. In: Bouajjani, A., Maler, O. (eds.) CAV 2009. LNCS, vol. 5643, pp. 337–352. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02658-4_27
27. Henzinger, T.A., Mikeev, L., Mateescu, M., Wolf, V.: Hybrid numerical solution of the chemical master equation. In: Computational Methods in Systems Biology (CMSB), pp. 55–65. ACM (2010)
28. Hepp, B., Gupta, A., Khammash, M.: Adaptive hybrid simulations for multiscale stochastic reaction networks. J. Chem. Phys. **142**(3), 034118 (2015)
29. Karp, R.M., Miller, R.E.: Parallel program schemata. J. Comput. Syst. Sci. **3**(2), 147–195 (1969)
30. Katoen, J.-P., Klink, D., Leucker, M., Wolf, V.: Three-valued abstraction for continuous-time Markov chains. In: Damm, W., Hermanns, H. (eds.) CAV 2007. LNCS, vol. 4590, pp. 311–324. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73368-3_37

31. Kwiatkowska, M., Thachuk, C.: Probabilistic model checking for biology. Softw. Syst. Saf. **36**, 165 (2014)
32. Lakin, M.R., Parker, D., Cardelli, L., Kwiatkowska, M., Phillips, A.: Design and analysis of DNA strand displacement devices using probabilistic model checking. J. R. Soc. Interface **9**(72), 1470–1485 (2012)
33. Larsen, K.G., Skou, A.: Bisimulation through probabilistic testing. Inf. Comput. **94**(1), 1–28 (1991)
34. Madsen, C., Myers, C., Roehner, N., Winstead, C., Zhang, Z.: Utilizing stochastic model checking to analyze genetic circuits. In: Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), pp. 379–386. IEEE (2012)
35. Mateescu, M., Wolf, V., Didier, F., Henzinger, T.A.: Fast adaptive uniformization of the chemical master equation. IET Syst. Biol. **4**(6), 441–452 (2010)
36. Munsky, B., Khammash, M.: The finite state projection algorithm for the solution of the chemical master equation. J. Chem. Phys. **124**, 044104 (2006)
37. Murata, T.: Petri nets: properties, analysis and applications. Proc. IEEE **77**(4), 541–580 (1989)
38. Rao, C.V., Arkin, A.P.: Stochastic chemical kinetics and the quasi-steady-state assumption: application to the Gillespie algorithm. J. Chem. Phys. **118**(11), 4999–5010 (2003)
39. Salis, H., Kaznessis, Y.: Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. J. Chem. Phys. **122**(5), 054103 (2005)
40. Soloveichik, D., Seelig, G., Winfree, E.: DNA as a universal substrate for chemical kinetics. Proc. Natl. Acad. Sci. U. S. A. **107**(12), 5393–5398 (2010)
41. Soudjani, S.E.Z., Abate, A.: Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes. SIAM J. Appl. Dyn. Syst. **12**(2), 921–956 (2013)
42. Esmaeil Zadeh Soudjani, S., Abate, A.: Precise approximations of the probability distribution of a Markov process in time: an application to probabilistic invariance. In: Ábrahám, E., Havelund, K. (eds.) TACAS 2014. LNCS, vol. 8413, pp. 547–561. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54862-8_45
43. Srivastava, R., You, L., Summers, J., Yin, J.: Stochastic vs. deterministic modeling of intracellular viral kinetics. J. Theor. Biol. **218**(3), 309–321 (2002)
44. Van Kampen, N.G.: Stochastic Processes in Physics and Chemistry, vol. 1. Elsevier, New York (1992)
45. Zhang, J., Watson, L.T., Cao, Y.: Adaptive aggregation method for the chemical master equation. Int. J. Comput. Biol. Drug Des. **2**(2), 134–148 (2009)