High-level stochastic simulation algorithm for exploring the dynamics of biological systems

1st George Assaf
Brandenburg University of Technology
Postbox 101344, 03013 Cottbus, Germany
george.assaf@b-tu.de

2nd Fei Liu
South China University of Technology
Guangzhou 510006, P.R. China
feiliu@scut.edu.cn

3rd Monika Heiner
Brandenburg University of Technology
Postbox 101344, 03013 Cottbus, Germany
monika.heiner@b-tu.de

***Abstract*—We propose a new algorithm for exploring behaviours of biological systems with repeated structures modelled as coloured stochastic Petri nets. The current approaches to execute coloured stochastic Petri net models over time require unfolding the model on hand into its uncoloured counterpart as a preliminary step before performing the actual stochastic simulation. In this context, unfolding coloured models with large scaling factors leads to memory explosion, as high-level modelled components need to be flattened into their basic reactions and associated species. Our proposed algorithm is based on incorporating the colouring concept into the well-known *Gillespie's* stochastic simulation algorithm. The proposed algorithm is implemented in the powerful Petri nets modelling and simulation tool *Snoopy*. In this paper, we present two biological case studies, for which the proposed algorithm reproduced identical results to the unfolding approach.**

***Index Terms*—biological systems with repeated structures, model analysis, stochastic simulation, coloured stochastic Petri nets**

I. Introduction

A. Modelling biological systems with repeated structures

Over the last decade, modelling and analysing biological systems with similar structures has become one of the focal points in the computational modelling community [1]–[4]. This interest comes from the fact that biological systems comprise thousands of similar biochemical reactions and associated species, that need to be modelled in a convenient manner for the purpose of predicting the behaviour of the modelled system. Coloured stochastic Petri nets ($\mathcal{SPN}^{\mathcal{C}}$) are an excelling modelling paradigm for obtaining quantitative insights into inherent stochasticity of the modelled system.

Petri nets ($\mathcal{PN}$) [5] have increasingly become an attractive formalism for modelling biological systems as they offer various graphical modelling elements in stochastic, deterministic, and hybrid paradigms for quantitatively analysing the behaviour of biological systems, see [6]–[8]. $\mathcal{PN}$ modelling elements basically comprise places, transitions, and directed arcs for connecting places and transitions. For biological systems, places model species whose molecule number/concentration (based on the modelling paradigm) is represented by the number of tokens (also called marking) that are held by the places, whereas transitions model the system reactions, and the connected arcs together with the associated arc weights represent the number of molecules of each involved reactant/product. Moreover, an occurrence of a reaction is equivalent to firing an enabled transition, for which all the transition's pre-places hold a number of tokens greater than or equal to the connected arc weights. Firing a transition results in consuming a number of tokens (from pre-places) equal to the corresponding arc weights, while the post-places are updated by adding tokens equal to the connected arc weights. To gain further knowledge of adopting $\mathcal{PN}$ for modellling and simulating biological systems, we recommend to consult, e.g. [9], [10].

Based on the traditional $\mathcal{PN}$, coloured Petri nets ($\mathcal{PN}^{\mathcal{C}}$) [11]–[13] were introduced to offer the concept of colour sets. A colour set is a user-defined data type with a finite number of elements (colours) that can be assigned to the net places to represent several copies of the modelled object. This means that each token will be differentiated by a unique colour, which has the advantage of representing similar objects using one coloured place rather than having serveral places per each object. It is worth mentioning that colour sets are also used to define user declarations such as variables and functions in the same way as programming languages which can be used later for annotating net places, transitions, and arcs. This has big advantages in expressing complex systems in a compact and scaleable fashion. Like plain (uncoloured) Petri nets, $\mathcal{PN}^{\mathcal{C}}$ offer quantitative classes for exploring the behaviour of biological systems with similar structures, see for example [2], [4], [14].

In this paper, we are interested in analysing the behaviours

George Assaf was supported by the *Katholischer Akademischer Ausländer-Dienst* (KAAD) scholarship. Fei Liu is supported by the National Natural Science Foundation of China (xxxx). Monika Heiner is an emeritus professor with a strong background in Petri nets theory and modelling and analysing techniques.

of biological systems with similar structures modelled using coloured stochastic Petri nets. Stochastic simulation is able to capture a realistic representation of random fluctuations at the molecular level and its effects on the behaviour of the modelled system. Interestingly, we propose a novel algorithm for analysing biological systems with similar structures modelled as $\mathcal{SPN}^{\mathcal{C}}$, as former approaches require unfolding the model on hand into its plain counterpart as a preliminary step before performing the actual simulation [15]. However, model unfolding results in memory space consumption, particularly for $\mathcal{SPN}^{\mathcal{C}}$ models with large scaling factors [1], [15]–[17]. Moreover, the unfolded model has to be fed to the simulator which will lead to additional overhead on both memory consumption and computational resources.

B. Coloured stochastic Petri nets

Coloured stochastic Petri nets [12], [18], [19] are the coloured version of the quantitative stochastic Petri nets ($\mathcal{SPN}$) [20], [21]. $\mathcal{SPN}^{\mathcal{C}}$ are an excellent tool for modelling and analysing biological systems with similar structures by capturing the system behaviour that is not observable in deterministic approaches by considering random interactions and fluctuations that are induced by the modelled system, and thus giving a better understanding of the underlying behaviour of biological phenomena, see for example [1], [2], [18], [19].

In plain $\mathcal{SPN}$, each place holds a discrete number of tokens to describe a specific state of the modelled object, e.g. a species. Each stochastic transition enjoys a stochastic firing rate function determining a stochastic waiting time before the actual firing takes place. Similar to $\mathcal{SPN}$, $\mathcal{SPN}^{\mathcal{C}}$ comprise discrete places and stochastic transitions. Additionally, each discrete place gets assigned a colour set permitting to specify a multi-set colour expression as place marking. The multi-set colour expression comprises a set of elements (colours) with their corresponding occurrences. Assuming the colour set (with three colours) CS ={a, b, c}, the multi-set expression {a‘1++b‘3++c‘5} describes that the colour a has one occurrence, the colour b has two occurrences and the colour c has 5 occurrences. It is worth mentioning that multi-set expressions are used to decorate arcs connecting coloured places and transitions for the purpose of determining the coloured token that will flow over arcs. Additional to stochastic rate functions, each coloured stochastic transition gets assigned a guard, which is a Boolean expression (constraint) determining the colour bindings, for which the corresponding enabled transition will fire. In the colour world, a transition gets enabled if the following condition holds.

$$\forall p \in {}^\bullet t, m(p) \geq f(p,t)\langle B \rangle, \tag{1}$$

where ${}^\bullet t$ denotes the pre-places of the transition t, $f(p,t)$ denotes the multi-set colour expression of the arc connecting the place p to the transition t. $\langle B \rangle$ denotes a binding set. The binding set is the colour combination of the variables that are involved in the colour expressions of the transition's adjacent arcs, for which the transition's guard expression is evaluated to *true* [22]. The condition given in Equation 1 describes that the transition t is said to be enabled using each binding $b(t) \in B$, if all the pre-places of t hold a number of coloured tokens greater than or equal to the corresponding evaluated colours (over arcs). In the following, we give the formal definition of $\mathcal{SPN}^{\mathcal{C}}$.

Definition 1 (Coloured stochastic Petri net [12]): A *coloured stochastic Petri net* is a 9-tuple $N = \langle P, T, A, \Sigma, c, g, f, v, m_0 \rangle$, where:

- P is a finite, non-empty set of places.
- T is a finite, non-empty set of transitions.
- $P \cap T = \emptyset$
- $A \subseteq (P \times T) \cup (T \times P)$ is a finite set of directed arcs.
- Σ is a finite, non-empty set of colour sets.
- $c : P \rightarrow \Sigma$ is a colour function that assigns to each place $p \in P$ a colour set $c(p) \in \Sigma$.
- $g : T \rightarrow EXP$ is a guard function that assigns to each transition $t \in T$ a guard expression of the Boolean type.
- $f : A \rightarrow EXP$ is an arc function that assigns to each arc $a \in A$ an arc expression of a multiset type $c(p)_{MS}$, where p is the place adjacent to the arc a.
- v: is a function that assigns a stochastic rate function to each transition $t \in T$.
- $m_0 : P \rightarrow EXP$ is an initialisation function that assigns to each place $p \in P$ an initialisation expression of a multiset type $c(p)_{MS}$.

In the following, we give an $\mathcal{SPN}^{\mathcal{C}}$ model of the scaleable repressilator as a running example for the purpose of illustrating the $\mathcal{SPN}^{\mathcal{C}}$ formal definition. The repressilator [23], [24] is a synthetic genetic network serving as a model to study gene regulatory networks and the dynamics of genetic oscillations. Furthermore, the repressilator is scaleable by the number of genes involved in the genetic network. Each gene produces a repressor protein that inhibits the expression of the next gene in a cyclic way. The produced protein can also be degraded. Note that each reaction in the system occurs with Mass-Action kinetics rate functions[a], see Table I.

TABLE I
RATE FUNCTIONS OF THE SYSTEM'S REACTIONS

reaction	*Rate function*	*Kinetic parameter*
generate	$k\_gen \times gene$	k_gen
block	$k\_block \times protein \times gene$	k_block
degrade	$k\_deg \times protein$	k_deg
unblock	$k\_unblock \times blocked$	$k\_unblock$

[a]MassAction(k) = k $\prod power(p, w)$ with $p \in {}^\bullet t$

Figure 1 gives a coloured stochastic Petri net model of the repressilator. First, the colour set *Gene* with *enum GeneSet* = {*a, b, c*} is defined, where each colour represents one gene in the system. Next, one variable of type *GeneSet*, e.g. x is declared to encode the active gene in the network. Then, the model is constructed by adding three coloured places that get assigned the colour set *GeneSet*. Each place represents system states that are either, gene, protein or blocked. The place *gene* is initialised with one token of each colour to describe that

the system starts with three genes with one molecule each. This is reflected in the model by assigning the colour function $1`all()$ to the place *gene*, determining one token of each colour yielding three coloured tokens. Moreover, all reactions that share the copies of reactants/products, e.g. genes/proteins are represented using one coloured stochastic transition whose rate function gets assigned according to Table I. For instance, the colour transition $generate$ is a high-level transition modelling three reactions behind, as the colour set *GeneSet* is assigned to its pre-place and post-place and this colour set comprises three colours. It is worth mentioning that net transitions do not get guard expressions, which means there is no colour constraint on enabling this transition. Afterwards, standard arcs (directed arcs) decorated with colour expressions are connected to express the reaction structures.

The big advantage of colouring is that the size of the modelled system can be adjusted by changing the colour sets of the model. For the repressilator, changing the colour set *GeneSet* by, e.g. adding more colours will scale the circular gene network by the newly added colour without having to touch the net structure, i.e. the net places/transitions and their connectivity style.

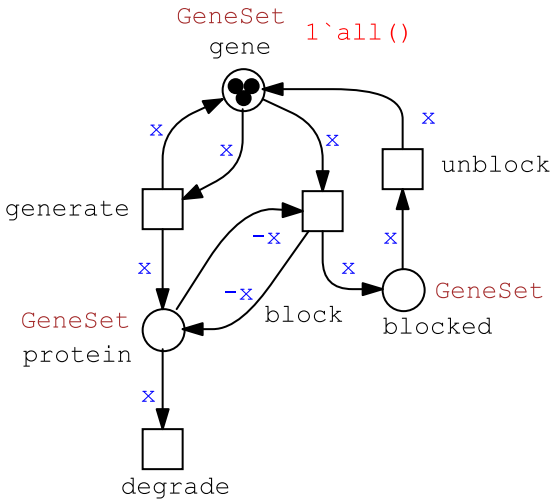


Fig. 1. $\mathcal{SPN}^{\mathcal{C}}$ of the scalable repressilator modelled using *Snoopy* [25]. The model can be scaled by adding new colours, i.e. gene Ids to the colour set *GeneSet*. Note that rate functions are not shown to keep the net readable.

II. Approach

A. Coloured stochastic Petri nets semantics

In order to present our high-level simulation algorithm, we are going to introduce the basic principles of simulating standard stochastic Petri nets. As uncoloured stochastic Petri nets, the underlying semantic of $\mathcal{SPN}^{\mathcal{C}}$ is defined by continuous-time Markov chain (CTMC) [26]. However, deriving the CTMC from $\mathcal{SPN}$ may be infeasible, as state space induced by the corresponding reachability graph can be infinite. Therefore, the CTMC has to be approximated for the purpose of simulating an $\mathcal{SPN}$ model. Building the approximated CTMC can be achieved by generating different paths through it. To this end, starting from the $\mathcal{SPN}$ initial marking, the net transitions have to be fired repeatedly [7]. To do so, two questions have to be answered. These questions are as follows. First, when the next transition will fire?. Second, which transition to fire?. Gillespie's stochastic simulation algorithm [27] (known as Gillespie's SSA algorithm) is adopted to answer these two questions.

According to the Gillespie's SSA, each reaction, i.e. a $\mathcal{PN}$ transition r_j defines a propensity function (also known as hazard function) $p_j(r)$ with the parameter k_j representing *mass-action* semantic, compare Equation 2

$$p_j(r) = k_j \prod_{i=1;p_i \in {}^\bullet t}^{N_j} \binom{m_{(p_i)}}{a_{ji}} \tag{2}$$

With m_i is the number of molecules of one reactant (a pre-place) of r_j and a_{ji} gives the corresponding stoichiometry. Then, the total propensity is calculated over all reactions using Equation 3.

$$TP_0(r) = \sum_{j=1}^{M} p_j(r) \tag{3}$$

In order to determine the next time step at which the next reaction will occur as well as to choose the reaction to occur, the algorithm has to generate two random numbers r_1 and r_2 according to the uniform distribution on the interval (0,1). The next time step is calculated using Equation 4:

$$N_t = -\frac{1}{TP_0(r)} ln r_1 \tag{4}$$

Afterwards, a reaction to occur R_μ is chosen according to Equation 5

$$\sum_{j=1}^{\mu-1} p_j(r) < r_2 TP_0(r) \leq \sum_{j=1}^{\mu} p_j(r) \tag{5}$$

The detailed algorithm implementation of the $\mathcal{SPN}$ simulation based on the Gillespie's SSA can be found in [7].

The former steps are directly performed on uncoloured $\mathcal{SPN}$ as a system reactions and the relevant species are explicitly given by mapping them to stochastic transitions and discrete places. However, simulating biological systems modelled as $\mathcal{SPN}^{\mathcal{C}}$ is not trivial, as the give $\mathcal{SPN}^{\mathcal{C}}$ model has to be unfolded into the corresponding $\mathcal{SPN}$ counterpart. $\mathcal{SPN}^{\mathcal{C}}$ unfolding includes unfolding all the places and transitions. In this context, net unfolding is time- as well as space-consuming as it requires solving the constraint satisfaction problems (CSP) induced by transition guards. For example, unfolding our running example given in Figure 1 produces the $\mathcal{SPN}$ presented in Figure 2. More details on coloured Petri nets unfolding can be found in [15].

In the following, we introduce our new approach for simulating $\mathcal{SPN}^{\mathcal{C}}$ models without having to unfold the net as a preliminary step.

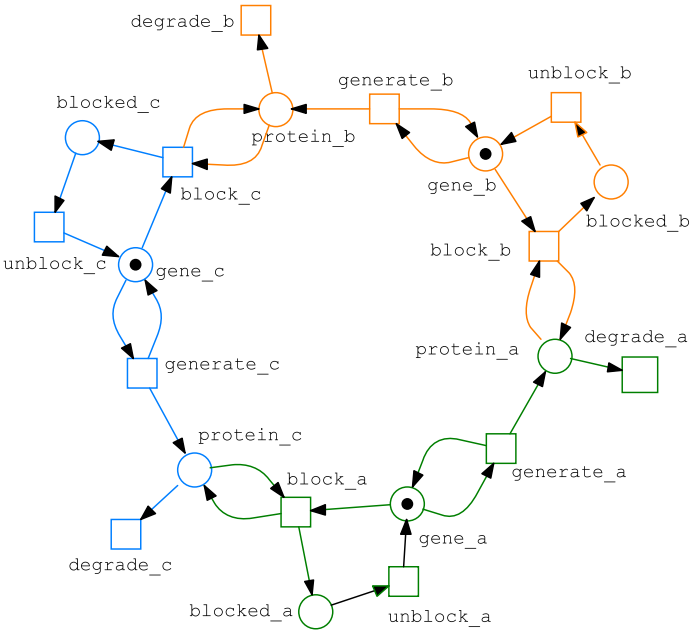


Fig. 2. Unfolded $\mathcal{SPN}$ model of the repressilator comprising 9 discrete places, 12 stochastic transitions, and 36 arcs. Net elements with the same colour belong to the same gene activities (component). The three gene activities have the same net structure.

B. High-level stochastic simulation algorithm

Our proposed solution for simulating $\mathcal{SPN}^{\mathcal{C}}$ models without the unfolding step is basically based on tuning Gillespie's SSA algorithm (see former Section) to work directly on the colour level. The proposed algorithm has to perform the following steps.

1) Simulator initialisation: includes writing the initial state of the $\mathcal{SPN}^{\mathcal{C}}$ model, i.e. model state at time $\tau = 0$ in the output trace file. In pursuit of this, the algorithm has to iterate over all the coloured places and then write the number of tokens of each colour (of every place) in the output trace file. Furthermore, the output trace file is initialised with the colour marking of every coloured place, i.e. the sum over all coloured tokens of every coloured place. It is worth mentioning that the evolution of coloured places is obtained by calculating the summation over all involved colours over time, which is an interesting measure for various biological systems. However, this measure is not considered by the standard stochastic simulation algorithm. Algorithm 1 sketches these steps.

Note that the algorithm formulate the output variable identifiers by concatenation of a place identifier with colour identifiers of the colours of the assigned colour set (line 6). Each resulting output variable is called a place instance. Moreover, the algorithm only will initialise output traces with a set of place instances given as input to the initialisation algorithm, so that only the variables of interest will be written instead of all place instances.

2) Determining the time at which the next reaction will occur: as in standard Gillespie, the algorithm has to determine the time step at which the next transition will occur. For this end, we introduce the colour propensity function, see Equation 6.

$$ {p_j}^b(r) = k_j \prod_{i=1;p_i \in {}^\bullet t_b}^{N_j} \binom{m_{(p_i)}}{{a_{ji}}^b} \tag{6} $$

Where

${p_j}^b(r)$ is the propensity function corresponding to one transition (reaction) whose binding is b (a transition instance),
${}^\bullet t_b$ gives the pre-places of the corresponding transition with the binding b,
$m_{(p_i)}$ gives the occurrence of the colour binding b at the coloured pre-place p_i of t,
${a_{ji}}^b$ gives the occurrence of colour binding b over the arcs connecting the coloured pre-place p_i of t.

Then, the total coloured propensity is obtained using Equation 7

$$ TCP_0(r) = \sum_{j=1}^{M} {p_j}^b(r) \tag{7} $$

For determining the next firing time, the algorithm has to follow the same principle of the standard Gillespie's SSA by generating two random numbers r_1 and r_2 according to the uniform distribution. Then, the next firing time is obtained using Equation 8

$$ N_{tc} = -\frac{1}{TCP_0(r)} lnr_1 \tag{8} $$

3) Determining next reaction to occur: choosing a reaction (a colour transition) to occur is equivalent to choosing a colour transition t whose index is μ together with one binding b, for which the transition t is enabled. To obtain these two pieces

Algorithm 1 Initialise output traces

Input: $\mathcal{SPN}^{\mathcal{C}}$ model with its initial state $m(\tau_0)$, S set of selected output variables.
Output: Output traces of place instance at time τ_0.

```
1:  for each coloured place p ∈ P do
2:      if p ∈ S then
3:          write overall marking of p;
4:      end if
5:      for each colour c ∈ C(p) do
6:          write place instance id such that concat(placeId(p),
        "_", colourId(c));
7:          if placeinstanceId ∈ S then
8:              if colour c ∈ m(p) then
9:                  write the occurrence of c as initial state;
10:             else
11:                 write 0 as initial marking;
12:             end if
13:         end if
14:     end for
15: end for
```

of information, Equation 5 needs to be adjusted as given in Equation 9.

$$\sum_{j=1}^{\mu-1} {p_j}^b(r) < r_2 TCP_0(r) \leq \sum_{j=1}^{\mu} {p_j}^b(r) \tag{9}$$

4) Removing isolated places: isolated places [15] are the places that are not directly reachable from the initial marking of a Petri net. Such places may be generated as there is no variable binding enabling a certain coloued transition due to, e.g., unfulfilled guard. This means that these places do not contribute to the model behaviour, i.e model execution over time. Therefore, these places can be safely excluded from the output trace file. Algorithm 2 cleans the output traces from the isolated places if they exist.

Algorithm 2 Removing isolated places.

Input: $\mathcal{SPN}^{\mathcal{C}}$ output traces.

Output: Output traces over time excluding isolated places.

```
1: for each instance place do
2:     if no evolution occurs over time then
3:         get rid of the corresponding column;
4:     end if
5: end for
```

Algorithm 3 sketches the required steps for simulating $\mathcal{SPN}^{\mathcal{C}}$ models. The algorithm takes a $\mathcal{SPN}^{\mathcal{C}}$ model together with the desired simulation end time, and produces model states over time as output. First, initialising the simulator is performed at the beginning (lines 1-2). This includes setting simulation time (τ) and writing the initial state of the model to the output trace file using Algorithm 1. Then, the simulator loop (lines 3 -18) has to be performed until the simulator time reaches the end time (τ_{end}). Afterwards, the simulation loop is started by iterating over coloured transitions. For each enabled coloured transition, the algorithm computes both colour propensity function as well as the total colour propensity using Equations 6 and 7 (lines 7-8), respectively. After that, two random numbers are generated according to the uniform distribution (line 12). Then, the time step at which the next transition will occur is obtained by applying Equation 8 and the transition to occur together with corresponding binding is selected by applying Equation 9 (lines 13-14). At this point, the chosen coloured transition t is fired using the binding b (line 15). as a result, the simulator clock is advanced by the obtained next time step (line 16) and the system state together with output traces are updated accordingly (line 17). Finally, isolated places are removed by performing Algorithm 2 (line 19).

It is worth mentioning that simulation traces can be reproduced by seeding the utilised random number generator by a seed value [28]. This feature has its advantages for verifying the correctness of our proposed algorithm by comparing the output traces with those generated by simulating the corresponding unfolded model using the standard Gillespie's SSA algorithm.

Algorithm 3 High-level stochastic simulation

Input: $\mathcal{SPN}^{\mathcal{C}}$ model with its initial state $m(\tau_0)$, simulation end time τ_{end}, *seed value*.

Output: Output traces of model variables (places).

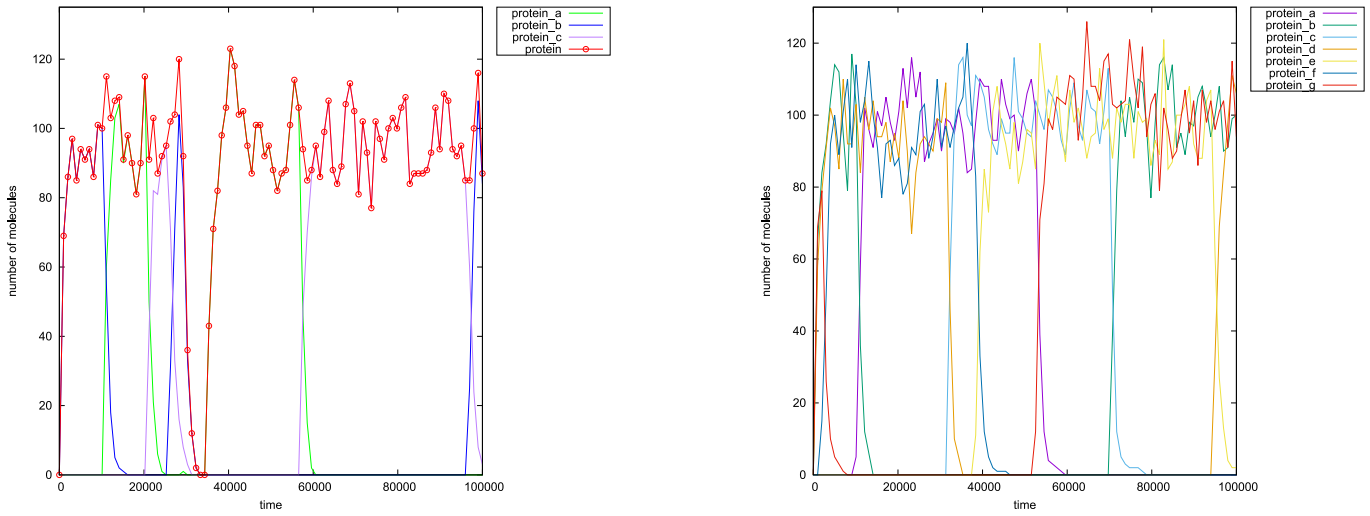
```
1: initialise simulator clock τ = 0.0;
2: initialise output traces using Algorithm 1;
3: while τ < τ_end do
4:     for each coloured transition t ∈ T do
5:         if t is enable then
6:             for each binding b ∈ B do
7:                 obtain colour propensity using Equation 6;
8:                 compute total propensity using Equation 7;
9:             end for
10:        end if
11:    end for
12:    generate two random variables r_1 and r_2 using seed;
13:    compute time step N_tc using Equation 8;
14:    select colour transition with one binding using Equation 9;
15:    fire colour transition t using binding b;
16:    advance simulator clock τ = τ + N_tc;
17:    update system state together with output traces;
18: end while
19: remove isolated places using Algorithm 2;
```

III. Another case study

In this section, we present another biological system, namely the *circadian rhythm* [29]. Circadian rhythm is an internal timing system that regulates various physiological and behavioural processes, e.g. the sleep/wake cycle in living organisms. The system comprises a set of genes (clock genes) that regulate their own transcription and translation in a cycle that takes approximately 24 h to complete [29], [30]. Assuming a system of two clock genes, e.g. *A, R*, gene transcription yields an *mRNA* whose translation produces a protein. While the first gene positively regulates its transcription, the second gene negatively regulates the corresponding transcription.

Figure III presents an $\mathcal{SPN}^{\mathcal{C}}$ model of the circadian rhythms. To encode to types of clock genes, we declare a colour set of *enum* type with two colours, i.e. *enum Genes* ={A, R}. Further, one variable of the type *Genes* is declared, i.e., *x : Genes*. Then, the coloured net is obtained by folding (colouring) the basic stochastic Petri net model introduced in [12]. The transcription of each gene (modelled by place G) is expressed by the transition r_1 which gives as a consequence an $mRNA$ molecule (modelled by place M), which in turn either degraded (r_4) or translated into a protein (place P). To learn more about modelling this sequence of biological activities together with the required modelling means, we recommend to consult [31].

In contrast to the running example (repressilator), reaction rates of the *circadian rhythm* are colour-dependent, which



(a) (b)

Fig. 3. Stochastic simulation traces of the repressilator $\mathcal{SPN}^{\mathcal{C}}$ model. For reproducibility purposes, random numbers have been seeded with the value 12345. (a) Scaleable $\mathcal{SPN}^{\mathcal{C}}$ model with 3 genes. (b) Scaleable $\mathcal{SPN}^{\mathcal{C}}$ model with 7 genes, namely *a,b,c,d,e,f,g*.

TABLE II
RATE FUNCTIONS OF THE CIRCADIAN RHYTHM. THE COLUMN $Gene$ REPRESENTS COLOUR DEPENDENCY OF RATE FUNCTIONS.

reaction	*Gene*	*Rate function*	*Kinetic parameter*
r1	A	$G \times aa_0$	aa_0
	R	$G \times ar_0$	ar_0
r2	A	$GP \times aa_1$	aa_1
	R	$GP \times ar_1$	ar_1
r3	A	$M \times ba$	ba
	R	$M \times br$	br
r4	A	$M \times dam$	dam
	R	$M \times drm$	drm
r5	A	$P \times da$	da
	R	$P \times dr$	dr
r6	A, R	$C \times da$	da
r7	A, R	$P \times gar$	gar
r8	A	$P \times G \times ga$	ga
	R	$P \times G \times gr$	gr
r9	A	$GP \times ta$	ta
	R	$GP \times tr$	tr

means the reaction, e.g. transcription of one gene occurs with a different rate constant than the transcription of the other gene. For instance, the reaction r_1 has colour-dependent rates, describing that the transcription of the gene A occurs with rate constant $aa0$, whereas the transcription of gene R occurs with rate constant $ar0$. Note that not all rate functions are shown in the $\mathcal{SPN}^{\mathcal{C}}$ model given in Figure III for having a better layout.

Figure III presents gives stochastic simulation traces of both mRNA and protein of the gene A, namely M_A and p_A, respectively.

IV. DISCUSSION

The proposed algorithm has shown that exploring behaviours of biological systems with similar structures can be achieved on the fly without having to flatten the modelled system into its basic reactions. The big advantages that are offered by our algorithm are as follows. First, there is no need to unfold coloured places, transitions and arcs to their uncoloured counterparts, and thus saving memory space. In this context, our algorithm offers to choose the variables of interest, i.e. places to be recorded in the simulation trace file. Second, our algorithm introduces a new measure of recording the evolution of high-level variables, i.e. coloured places over time which is not offered by the standard stochastic simulation algorithm.

For the purpose of evaluating the performance of our algorithm, we performed both standard stochastic simulation algorithm and high-level simulation algorithm implemented in *Snoopy* [25] on the presented running example (the repressilator). Table III gives the facts representing time spent by both algorithms against different settings of simulation. Note that the random number generator has been seeded with a random seed number per each experiment[b] .

Table III shows that the standard Gillespie's stochastic simulation algorithm outperforms the current implementation of the high-level stochastic simulation algorithm by its run time In both standard simulation and high-level simulation the time is spent on calculating the next time point at which the next transition will fire. This piece of information is obtained by calculating the propensity functions of all enabled (coloured) reactions, i.e. enabled transitions per every iteration of the simulator loop. In this context, firing one transition per iteration will have only influence on the those transition that are directly connected with the pre/post-places of the fired one. However, *Snoopy's* implementation of the standard simulation algorithm addresses this matter by only calculating the propensity functions of those transitions that have been affected by the previous transition's firing [7]. The current implementation of the high-level simulation algorithm does not consider this issue so far, as it will be subject for further work (see next Section).

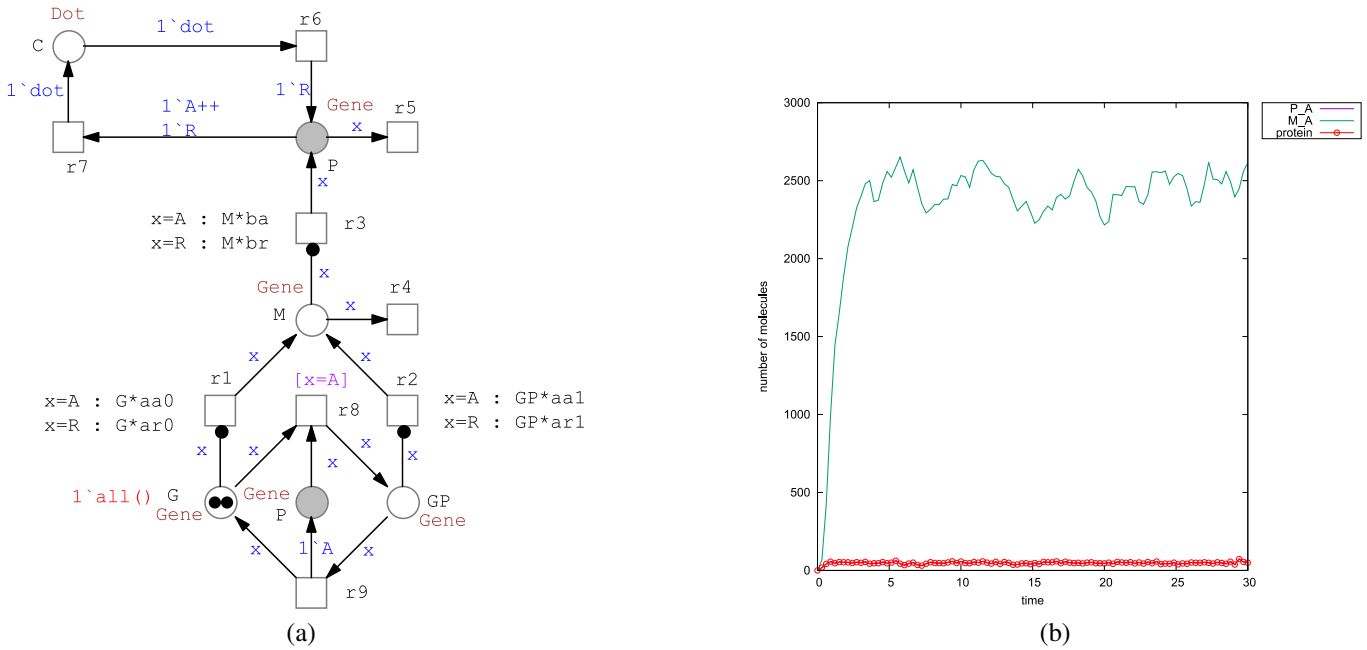


Fig. 4. (a) Coloured stochastic Petri net of the circadian rhythm system, in which system is initialised with one molecule of each gene. Not that not all rate functions are shown in the model for readability purposes. (b) Stochastic simulation traces of the *mRNA* and protein produced by gene A. For reproducibility purposes, random numbers have been seeded with the value 1624402775.

TABLE III
PERFORMANCE EVALUATION

End time (time unit)	High-level simulation	Standard simulation
50	1.24 sec	$<$ 1 sec
100	3.73 sec	$<$ 1 sec
500	6.75 sec	$<$ 1 sec
1000	21.93 sec	$<$ 1 sec
2000	38.95 sec	$<$ 1 sec
10 000	3 m 21 sec	$<$ 1 sec
100 000	26 m 7 sec	1.01 sec

[b]Experiments have been performed on Windows machine with Core i7 CPU/1.80 GHz, RAM capacity of 32 GB.

V. Conclusion

In this paper, we introduced a new stochastic simulation algorithm for analysing the behaviour of biological systems characterised by having components with their similar structures. Our proposed algorithm is based on adapting the well-known Gillespie's stochastic simulation algorithm without having to unfold the modelled system, which has basically a big advantage of saving memory space.

Furthermore, we presented two coloured stochastic Petri net models of two biological case studies. Performing the high-level stochastic simulation algorithm reproduced identical results to the standard stochastic simulation algorithm. For reproducibility purposes, our presented models can be accessed via [32]. The latest deployed version of *Snoopy* [25] is accessible via [33]. Moreover, supplementary material can be accessed via [32] for guiding interested readers on performing high-level simulation in *Snoopy*.

Our future work includes improving the efficiency of our algorithm by constructing the dependency graph of the coloured net as discussed in Section IV. Moreover, improving user support by allowing to write of formula for transitions' rate functions, e.g. *MassAction(k)* instead of writing the pre-places explicitly as it supports tool usability. Furthermore, for biological systems, it is interesting to gain knowledge on the number of event occurrences, i.e. transition firings and their schedule as the simulator evolves. This also will be in the course of our future work package.

Acknowledgment

We would like to thank David Gilbert for the fruitful discussions.

References

[1] O. Pârvu, D. Gilbert, M. Heiner, F. Liu, N. Saunders, and S. Shaw, "Spatial-temporal modelling and analysis of bacterial colonies with phase variable genes," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 25, no. 2, p. 25p., May 2015.

[2] F. Liu, M. Heiner, and D. Gilbert, "Coloured Petri nets for multilevel, multiscale, and multidimensional modelling of biological systems," *Briefings in Bioinformatics*, vol. 20, no. 3, pp. 877–886, 2019.

[3] A. Ismail, M. Herajy, E. Atlam, and M. Heiner, "A Graphical Approach for Hybrid Simulation of 3D Diffusion Bio-models via Coloured Hybrid Petri Nets," *Modelling and Simulation in Engineering*, vol. 2020, 2020.

[4] S. Connolly, D. Gilbert, and M. Heiner, "From Epidemic to Pandemic Modelling," *Frontiers in Systems Biology*, March 2022.

[5] W. Reisig, "Petri nets and algebraic specifications," *Theoretical Computer Science*, vol. 80, no. 1, pp. 1–34, 1991.

[6] M. Herajy, "Computational steering of multi-scale biochemical networks," Ph.D. dissertation, BTU Cottbus, Dep. of CS, January 2013.

[7] C. Rohr, "Simulative analysis of coloured extended stochastic petri nets," Ph.D. dissertation, BTU Cottbus, Dep. of CS, January 2017.

[8] M. Herajy and M. Heiner, "Adaptive and Bio-semantics of Continuous Petri Nets: Choosing the Appropriate Interpretation," *Fundamenta Informaticae*, vol. 160, no. 1-2, pp. 53–80, 2018.

[9] M. Blätke, C. Rohr, M. Heiner, and W. Marwan, *A Petri Net based Framework for Biomodel Engineering*, ser. Modeling and Simulation in Science, Engineering and Technology. Springer, Birkhäuser Mathematics, December 2014, pp. 317–366.

[10] M. Blätke, M. Heiner, and W. Marwan, *BioModel Engineering with Petri Nets*. Elsevier Inc., March 2015, ch. 7, pp. 141–193.

[11] K. Jensen, "Coloured petri nets," in *Petri Nets: Central Models and Their Properties*, W. Brauer, W. Reisig, and G. Rozenberg, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987.

[12] F. Liu, "Colored petri nets for systems biology," Ph.D. dissertation, BTU Cottbus, Dep. of CS, 2012.

[13] G. Assaf, " Fuzzy coloured Petri nets for modelling biological systems with uncertain kinetic parameters ," Ph.D. dissertation, BTU Cottbus, Dep. of CS, 2022.

[14] M. H. A Ismail and M. Heiner, *A Graphical Approach for the Hybrid Modelling of Intracellular Calcium Dynamics Based on Coloured Hybrid Petri Nets*, ser. Computational Biology. Springer, Cham, 2019, vol. 30, ch. 13, pp. 349–367.

[15] M. Schwarick, C. Rohr, F. Liu, G. Assaf, J. Chodak, and M. Heiner, "Efficient unfolding of coloured petri nets using interval decision diagrams," in *Proc. PETRI NETS 2020*, ser. LNCS, T. C. R Janicki, N Sidorova, Ed., vol. 12152. Springer, Cham, June 2020, pp. 324–344.

[16] Q. Gao, D. Gilbert, M. Heiner, F. Liu, D. Maccagnola, and D. Tree, "Multiscale Modelling and Analysis of Planar Cell Polarity in the Drosophila Wing," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 2, pp. 337–351, 2013.

[17] F. Liu, M. Blätke, M. Heiner, and M. Yang, "Modelling and simulating reaction–diffusion systems using coloured petri nets," *Computers in Biology and Medicine*, vol. 53, pp. 297–308, 2014.

[18] F. Liu and M. Heiner, "Modeling membrane systems using colored stochastic petri nets," *Nat. Computing*, vol. 12, no. 4, pp. 617 – 629, 2013.

[19] ——, "Multiscale modelling of coupled ca2+ channels using coloured stochastic petri nets," *IET Systems Biology*, vol. 7, no. 4, pp. 106 – 113, August 2013.

[20] M. Heiner, S. Lehrack, D. Gilbert, and W. Marwan, "Extended stochastic petri nets for model-based design of wetlab experiments," *Transactions on Computational Systems Biology XI*, vol. 5750, pp. 138–163, 2009.

[21] C. Rohr, "Discrete-Time Leap Method For Stochastic Simulation," *Fundamenta Informaticae*, vol. 160, no. 1-2, pp. 181–198, 2018.

[22] L. M. Kristensen and S. Christensen, "Implementing coloured petri nets using a functional programming language," *Higher-Order and Symbolic Computation*, vol. 17, no. 3, pp. 207–243, Sep 2004.

[23] M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators," *Nature*, vol. 403, no. 6767, pp. 335–338, Jan 2000.

[24] F. Liu and M. Heiner, *Petri Nets for Modeling and Analyzing Biochemical Reaction Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 245–272.

[25] M. Heiner, M. Herajy, F. Liu, C. Rohr, and M. Schwarick, "Snoopy – a unifying petri net tool," in *Application and Theory of Petri Nets*, S. Haddad and L. Pomello, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 398–407.

[26] M. Heiner, D. Gilbert, and R. Donaldson, "Petri nets for systems and synthetic biology," in *Formal Methods for Computational Systems Biology*, M. Bernardo, P. Degano, and G. Zavattaro, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 215–264.

[27] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *Journal of Computational Physics*, vol. 22, no. 4, pp. 403–434, 1976.

[28] J. Chodak, "Spike – a tool for reproducible simulation experiments," Ph.D. dissertation, BTU Cottbus, Dep. of CS, December 2022.

[29] D. Gonze, J. Halloy, and A. Goldbeter, "Deterministic versus stochastic models for circadian rhythms," *Journal of Biological Physics*, vol. 28, pp. 637–653, Dec 2002.

[30] S. Hood and S. Amir, "Biological clocks and rhythms of anger and aggression," *Frontiers in Behavioral Neuroscience*, vol. 12, 2018. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnbeh.2018.00004

[31] F. Liu, G. Assaf, M. Chen, and M. Heiner, "A Petri nets-based framework for whole-cell modeling," *BioSystems*, vol. 210, p. 104533, 2021.

[32] (2023) Coloured stochastic petri net models. [Online]. Available: https://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Examples

[33] (2023) Snoopy - petri nets modelling and simulation tool. [Online]. Available: https://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Snoopy