# Reducing Spreading Processes on Networks to Markov Population Models

Gerrit Großmann[1(✉)] and Luca Bortolussi[1,2]

[1] Saarland University, 66123 Saarbrücken, Germany
gerrit.grossmann@uni-saarland.de
[2] University of Trieste, Trieste, Italy
lbortolussi@units.it

**Abstract.** Stochastic processes on complex networks, where each node is in one of several compartments, and neighboring nodes interact with each other, can be used to describe a variety of real-world spreading phenomena. However, computational analysis of such processes is hindered by the enormous size of their underlying state space.

In this work, we demonstrate that lumping can be used to reduce any epidemic model to a Markov Population Model (MPM). Therefore, we propose a novel lumping scheme based on a partitioning of the nodes. By imposing different types of counting abstractions, we obtain coarse-grained Markov models with a natural MPM representation that approximate the original systems. This makes it possible to transfer the rich pool of approximation techniques developed for MPMs to the computational analysis of complex networks' dynamics.

We present numerical examples to investigate the relationship between the accuracy of the MPMs, the size of the lumped state space, and the type of counting abstraction.

**Keywords:** Epidemic modeling · Markov Population Model · Lumping · Model reduction · Spreading process · SIS model · Complex networks

## 1 Introduction

Computational modeling and analysis of dynamic processes on networked systems is a wide-spread and thriving research area. In particular, much effort has been put into the study of spreading phenomena [2,16,28,38]. Arguably, the most common formalism for spreading processes is the so-called Susceptible-Infected-Susceptible (SIS) model with its variations [28,38,39].

In the SIS model, each node is either *infected* (I) or *susceptible* (S). Infected nodes propagate their infection to neighboring susceptible nodes and become susceptible again after a random waiting time. Naturally, one can extend the number of possible node states (or compartments) of a node. For instance, the SIR model introduces an additional *recovered* state in which nodes are immune to the infection.

D. Parker and V. Wolf (Eds.): QEST 2019, LNCS 11785, pp. 292–309, 2019.
https://doi.org/10.1007/978-3-030-30281-8_17

SIS-type models are remarkable because—despite their simplicity—they allow the emergence of complex macroscopic phenomena guided by the topological properties of the network. There exists a wide variety of scenarios which can be described using the SIS-type formalism. For instance, the SIS model has been successfully used to study the spread of many different pathogens like influenza [26], dengue fever [40], and SARS [36]. Likewise, SIS-type models have shown to be extremely useful for analyzing and predicting the spread of opinions [29,49], rumors [52,53], and memes [51] in online social networks. Other areas of applications include the modeling of neural activity [15], the spread of computer viruses [11] as well as blackouts in financial institutions [34].

The semantics of SIS-type processes can be described using a continuous-time Markov chain (CTMC) [28,47] (cf. Sect. 3 for details). Each possible assignment of nodes to the two node states S and I constitutes an individual state in the CTMC (here referred to as *network state* to avoid confusion[1]). Hence, the CTMC state space grows exponentially with the number of nodes, which renders the numeral solution of the CTMC infeasible for most realistic contact networks.

This work investigates an aggregation scheme that *lumps* similar network states together and thereby reduces the size of the state space. More precisely, we first partition the nodes of the contact network. After which, we impose a counting abstraction on each partition. We only lump two networks states together when their corresponding counting abstractions coincide on each partition.

As we will see, the counting abstraction induces a natural representation of the lumped CTMC as a Markov Population Model (MPM). In an MPM, the CTMC states are vectors which, for different types of species, count the number of entities of each species. The dynamics can elegantly be represented as species interactions. More importantly, a very rich pool of approximation techniques has been developed on the basis of MPMs, which can now be applied to the lumped model. These include efficient simulation techniques [1,7], dynamic state space truncation [24,33], moment-closure approximations [19,44], linear noise approximation [18,46], and hybrid approaches [4,43].

The remainder of this work is organized as follows: Sect. 2 shortly revises related work, Sect. 3 formalized SIS-type models and their CTMC semantics. Our lumping scheme is developed in Sect. 4. In Sect. 5, we show that the lumped CTMCs have a natural MPM representation. Numerical results are demonstrated in Sect. 6 and some conclusions in Sect. 7 complete the paper and identify open research problems.

## 2 Related Work

The general idea behind *lumping* is to reduce the complexity of a system by aggregating (i.e., lumping) individual components of the system together. Lumping is a popular model reduction technique which has been used to reduce the number of equations in a system of ODEs and the number of states in a Markov

[1] In the following, we will use the term CTMC state and network state interchangeably.

chain, in particular in the context of biochemical reaction networks [6,8,31,50]. Generally speaking, one can distinguish between *exact* and *approximate* lumping [6,31].

Most work on the lumpability of epidemic models has been done in the context of exact lumping [28,42,48]. The general idea is typically to reduce the state space by identifying symmetries in the CTMC which themselves can be found using symmetries (i.e., automorphisms) in the contact network. Those methods, however, are limited in scope because these symmetries are infeasible to find in real-world networks and the state space reduction is not sufficient to make realistic models small enough to be solvable.

This work proposes an approximate lumping scheme. Approximate lumping has been shown to be useful when applied to mean-field approximation approaches of epidemic models like the degree-based mean-field and pair approximation equations [30], as well as the approximate master equation [14,21]. However, mean-field equations are essentially inflexible as they do not take topological properties into account or make unrealistic independence assumptions between neighboring nodes.

Moreover, [27] proposed using local symmetries in the contact network instead of automorphisms to construct a lumped Markov chain. This scheme seems promising, in particular on larger graphs where automorphisms often do not even exist, however, the limitations for real-world networks due to a limited amount of state space reduction and high computational costs seem to persist.

Conceptually similar to this work is also the *unified mean-field framework* (UMFF) proposed by Devriendt et al. in [10]. Devriendt et al. also partition the nodes of the contact network but directly derive a mean-field equation from it. In contrast, this work focuses on the analysis of the lumped CTMC and its relation to MPMs. Moreover, we investigate different types of counting abstractions, not only node based ones. The relationship between population dynamics and networks has also been investigated with regard to Markovian agents [3].

## 3 Spreading Processes

Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ be a an undirected graph without self-loops. At each time point $t \in \mathbb{R}_{\geq 0}$ each node occupies one of $m$ different node states, denoted by $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$ (typically, $\mathcal{S} = \{\mathtt{S}, \mathtt{I}\}$). Consequently, the network state is given by a labeling $x : \mathcal{N} \rightarrow \mathcal{S}$. We use

$$\mathcal{X} = \{x \mid x : \mathcal{N} \rightarrow \mathcal{S}\}$$

to denote all possible labelings. $\mathcal{X}$ is also the state space of the underlying CTMC. As each of the $|\mathcal{N}|$ nodes occupies one of $m$ states, we find that $|\mathcal{X}| = |\mathcal{S}|^{|\mathcal{N}|}$.

A set of stochastic rules determines the particular way in which nodes change their corresponding node states. Whether a rule can be applied to a node depends on the state of the node and of its immediate neighborhood.

The neighborhood of a node is modeled as a vector $\mathbf{m} \in \mathbb{Z}_{\geq 0}^{|\mathcal{S}|}$ where $\mathbf{m}[s]$ denotes the number of neighbors in state $s \in \mathcal{S}$ (we assume an implicit enumeration of states). Thus, the degree (number of neighbors, denoted by $k$) of

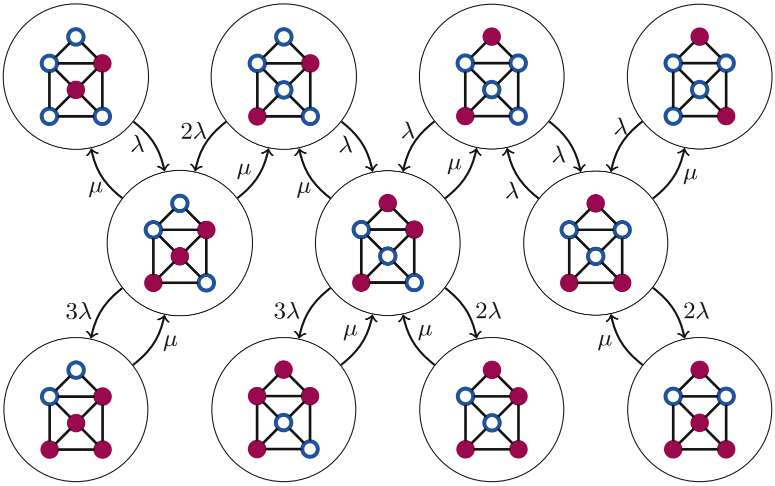


**Fig. 1.** The CTMC induced by the `SIS` model (`S`: *blue*, `I`: *magenta, filled*) on a toy graph. Only a subset of the CTMC spate space (11 out of $2^6 = 64$ network states) is shown. (Color figure online)

a node is equal to the sum over its associated neighborhood vector, that is, $k = \sum_{s \in \mathcal{S}} \mathbf{m}[s]$. The set of possible neighborhood vectors is denoted as

$$\mathcal{M} = \left\{ \mathbf{m} \in \mathbb{Z}_{\geq 0}^{|\mathcal{S}|} \;\middle|\; \sum_{s \in \mathcal{S}} \mathbf{m}[\mathrm{s}] \leq k_{\max} \right\},$$

where $k_{\max}$ denotes the maximal degree in a given network.

Each rule is a triplet $s_1 \xrightarrow{f} s_2$ $(s_1, s_2 \in \mathcal{S}, s_1 \neq s_2)$, which can be applied to each node in state $s_1$. When the rule "fires" it transforms the node from $s_1$ into $s_2$. The rate at which a rule "fires" is specified by the rate function $f : \mathcal{M} \rightarrow \mathbb{R}_{\geq 0}$ and depends on the node's neighborhood vector. The time delay until the rule is applied to the network state is drawn from an exponential distribution with rate $f(\mathbf{m})$. Hence, higher rates correspond to shorter waiting times. For the sake of simplicity and without loss of generality, we assume that for each pair of states $s_1$, $s_2$ there exists at most one rule that transforms $s_1$ to $s_2$.

In the well-known `SIS` model, infected nodes propagate their infection to susceptible neighbors. Thus, the rate at which a susceptible node becomes infected is proportional to its number of infected neighbors:

$$\mathtt{S} \xrightarrow{f} \mathtt{I} \qquad \text{with} \quad f(\mathbf{m}) = \lambda \cdot \mathbf{m}[\mathtt{I}],$$

where $\lambda \in \mathbb{R}_{\geq 0}$ is a rule-specific rate constant (called *infection rate*) and $\mathbf{m}[\mathtt{I}]$ denotes the number of infected neighbors. Furthermore, a recovery rule transforms infected nodes back to being susceptible:

$$\mathtt{I} \xrightarrow{f} \mathtt{S} \qquad \text{with} \quad f(\mathbf{m}) = \mu,$$

where $\mu \in \mathbb{R}_{\geq 0}$ is a rule-specific rate constant called *recovery rate*.

A variation of the SIS model is the SI model where no curing rule exists and all nodes (that are reachable from an infected node) will eventually end up being infected. Intuitively, each rule tries to "fire" at each position $n \in \mathcal{N}$ where it can be applied. The rule and node that have the shortest waiting time "win" and the rule is applied there. This process is repeated until some stopping criterion is fulfilled.

### 3.1 CTMC Semantics

Formally, the semantics of the SIS-type processes can be given in terms of continuous-time Markov Chains (CTMCs). The state space is the set of possible network states $\mathcal{X}$. The CTMC has a transition from state $x$ to $x'$ ($x, x' \in \mathcal{X}$, $x \neq x'$) if there exists a node $n \in \mathcal{N}$ and a rule $s_1 \xrightarrow{f} s_2$ such that the application of the rule to $n$ transforms the network state from $x$ to $x'$. The rate of the transition is exactly the rate $f(\mathbf{m})$ of the rule when applied to $n$. We use $q(x, x') \in \mathbb{R}_{\geq 0}$ to denote the transition rate between two network states. Figure 1 illustrates the CTMC corresponding to an SIS process on a small toy network.

Explicitly computing the evolution of the probability of $x \in \mathcal{X}$ over time with an ODE solver, using numerical integration, is only possible for very small contact networks, since the state space grows exponentially with the number of nodes. Alternative approaches include sampling the CTMC, which can be done reasonably efficiently even for comparably large networks [9,22,45] but is subject to statistical inaccuracies and is mostly used to estimate global properties.

## 4 Approximate Lumping

Our lumping scheme is composed of three basic ingredients:
**Node Partitioning:** The partitioning over the nodes $\mathcal{N}$ that is explicitly provided.
**Counting Pattern:** The type of features we are counting, i.e., nodes or edges.
**Implicit State Space Partitioning:** The CTMC state space is implicitly partitioned by counting the nodes or edges on each node partition.

We will start our presentation discussing the partitioning of the state space, then showing how to obtain it from a given node partitioning and counting pattern. To this end, we use $\mathcal{Y}$ to denote the new *lumped* state space and assume that there is a surjective[2] lumping function

$$\mathcal{L} : \mathcal{X} \rightarrow \mathcal{Y}$$

that defines which network states will be lumped together. Note that the lumped state space is the image of the lumping function and that all network states $x \in \mathcal{X}$ which are mapped to the same $y \in \mathcal{Y}$ will be aggregated.

[2] If $\mathcal{L}$ is not surjective, we consider only the image of $\mathcal{L}$ to be the lumped state space.

Later in this section, we will discuss concrete realizations of $\mathcal{L}$. In particular, we will construct $\mathcal{L}$ based on a node partitioning and a counting abstraction of our choice. Next, we define the transition rates $q(y, y')$ (where $y, y' \in \mathcal{Y}$, $y \neq y'$) between the states of the lumped Markov chain:

$$q(y, y') = \frac{1}{|\mathcal{L}^{-1}(y)|} \sum_{x \in \mathcal{L}^{-1}(y)} \sum_{x' \in \mathcal{L}^{-1}(y')} q(x, x')\,. \tag{1}$$

This is simply the mean transition rate at which an original state from $x$ goes to some $x' \in \mathcal{L}^{-1}(y')$. Technically, Eq. (1) corresponds to the following *lumping assumption*: we assume that at each point in time all network states belonging to a lumped state $y$ are equally likely.

### 4.1 Partition-Based Lumping

Next, we construct the lumping function $\mathcal{L}$. Because we want to make our lumping aware of the contact network's topology, we assume a given partitioning $\mathcal{P}$ over the nodes $\mathcal{N}$ of the contact network. That is, $\mathcal{P} \subset 2^{\mathcal{N}}$ and $\bigcup_{P \in \mathcal{P}} P = \mathcal{N}$ and all $P \in \mathcal{P}$ are disjoint and non-empty. Based on the node partitioning, we can now impose different kinds of counting abstractions on the network state. This work considers two types: counting nodes and counting edges. The counting abstractions are visualized in Fig. 3. A full example of how a lumped CTMC of an `SI` model is constructed using the node-based counting abstraction is given in Fig. 2.

**Node-Based Counting Abstraction.** We count the number of nodes in each state and partition. Thus, for a given network state $x \in \mathcal{X}$, we use $y(s, P)$ to denote the number of nodes in state $s \in \mathcal{S}$ in partition $P \in \mathcal{P}$. The lumping function $\mathcal{L}$ projects $x$ to the corresponding counting abstraction. Formally:

$$\begin{aligned} &\mathcal{Y} = \{y \mid y : \mathcal{S} \times \mathcal{P} \to \mathbb{Z}_{\geq 0}\} \\ &\mathcal{L}(x) = y \\ \text{with:}\quad &y(s, P) = |\{n \in \mathcal{N} \mid X(n) = s, n \in P\}|\,. \end{aligned}$$

**Edge-Based Counting Abstraction.** Again, we assume that a network state $x$ and a node partitioning $\mathcal{P}$ are given. Now we count the edges, that is for each pair of states $s, s' \in \mathcal{S}$ and each pair of partitions $P, P' \in \mathcal{P}$, we count $y(s, P, s', P')$ which is the number of edges $(n, n') \in \mathcal{E}$ where $x(n) = s$, $n \in P$, $x(n') = s'$, $n' \in P'$. Note that this includes cases where $P = P'$ and $s = s'$. However, only counting the edges does not determine how many nodes there are in each state (see Fig. 3 for an example).

In order to still have this information encoded in each lumped state, we slightly modify the network structure by adding a new dummy node $n_\star$ and connecting each node to it . The dummy node has a dummy state denoted by

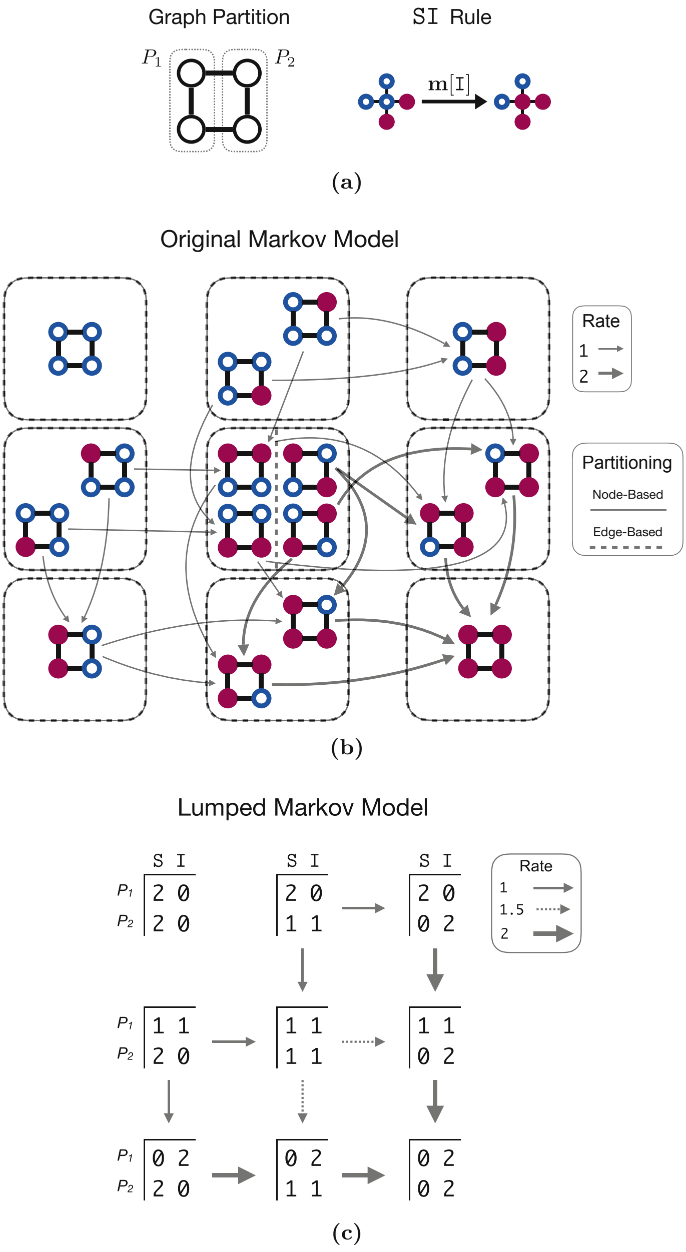
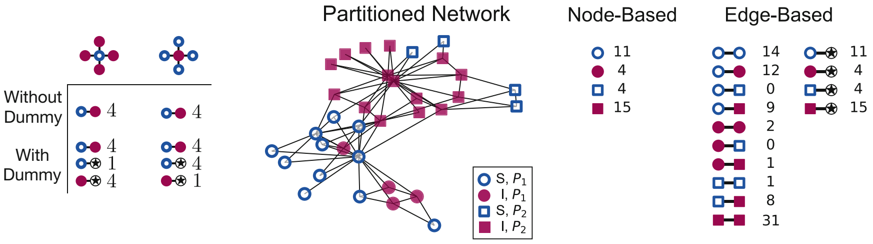


**Fig. 2.** Illustration of the lumping process. (a): Model. A basic SI-Process where infected nodes (magenta, filled) infect susceptible neighbors (blue) with rate infection $\lambda = 1$. The contact graph is divided into two partitions. (b): The underlying CTMC with $2^4 = 16$ states. The graph partition induces the edge-based and node-based lumping. The edge-based lumping refines the node-based lumping and generates one partition more (vertical line in the central partition). (c): The lumped CTMC using node-based counting abstraction with only 9 states. The rates are the averaged rates from the full CTMC. (Color figure online)



(a) (b)

**Fig. 3.** (a) By adding the dummy-node, the edge-based abstraction is able to differentiate the two graphs. Adding the dummy-node ensures that the nodes in each state are counted in the edge-based abstraction. (b) Left: A partitioned network (Zachary's Karate Club graph from [12]) (S: *blue*, I: *magenta, filled*). The network is partitioned into $P_1$ (○-nodes) and $P_2$ (□-nodes). Right: The corresponding counting abstractions. (Color figure online)

$\star$ which never changes, and it can be assigned to a new dummy partition $P_\star$. Formally,

$$\mathcal{N} := \mathcal{N} \cup \{n_\star\} \qquad \mathcal{S} := \mathcal{S} \cup \{\star\} \qquad L(n_\star) = \star \qquad \mathcal{P} := \mathcal{P} \cup \{P_\star\}$$
$$\mathcal{E} := \mathcal{E} \cup \{(n, n_\star) \mid n \in \mathcal{N}, n \neq n_\star\}.$$

Note that the rate function $f$ ignores the dummy node. The lumped representation is then given as:

$$\mathcal{Y} = \{y \mid y : \mathcal{S} \times \mathcal{P} \times \mathcal{S} \times \mathcal{P} \to \mathbb{Z}_{\geq 0}\}$$
$$\mathcal{L}(x) = y$$
$$\text{with:} \quad y(s, P, s', P') = \left|\left\{(n, n') \in \mathcal{E} \mid x(n) = s, n \in P, x(n') = s', n' \in P'\right\}\right|$$

**Example.** Figure 2 illustrates how a given partitioning and the node-based counting approach induces a lumped CTMC. The partitions induced by the edge-based counting abstracting are also shown. In this example, the edge-based lumping aggregates only isomorphic network states.

### 4.2 Graph Partitioning

Broadly speaking, we have three options to partition the nodes based on local features (e.g., its degree) or global features (e.g., communities in the graph) or randomly. As a baseline, we use a random node partitioning. Therefore, we fix the number of partitions and randomly assign each node to a partition while enforcing that all partitions have, as far as possible, the same number of elements.

Moreover, we investigate a degree-based partitioning, where we define the distance between to nodes $n, n'$ as their relative degree difference (similar to [30]):

$$d_{\mathrm{k}}(n,n') = \frac{|k_n - k_{n'}|}{\max(k_n, k_{n'})}\,.$$

We can then use any reasonable clustering algorithm and build partitions (i.e., clusters) with the distance function. In this work, we focus on bottom-up hierarchical clustering as it provides the most principled way of precisely controlling the number of partitions. Note that, for the sake of simplicity (in particular, to avoid infinite distances), we only consider contact networks where each node is reachable from every other node. We break ties arbitrarily.

To get a clustering considering global features we use a spectral embedding of the contract network. Specifically, we use the `spectral_layout` function from the `NetworkX` Python-package [23] with three dimensions and perform hierarchical clustering on the embedding. In future research, it would be interesting to compute node distances based on more sophisticated graph embedding as the ones proposed in [17]. Note that in the border cases $|\mathcal{P}| = 1$ and $|\mathcal{P}| = |\mathcal{N}|$ all methods yield the same partitioning.

## 5 Markov Population Models

Markov Population Models (MPMs) are a special form of CTMCs where each CTMC state is a population vector over a set of species. We use $\mathcal{Z}$ to denote the finite set of species (again, with an implicit enumeration) and $\mathbf{y} \in \mathbb{Z}_{\geq 0}^{|\mathcal{Z}|}$ to denote the population vector. Hence, $\mathbf{y}[z]$ identifies the number of entities of species $z$. The stochastic dynamics of MPMs is typically expressed as a set of reactions $\mathcal{R}$, each reaction, $(\alpha, \mathbf{b}) \in \mathcal{R}$, is comprised of a propensity function $\alpha : \mathbb{Z}_{\geq 0}^{|\mathcal{Z}|} \to \mathbb{R}_{\geq 0}$ and a change vector $\mathbf{b} \in \mathbb{Z}^{|\mathcal{Z}|}$. When reaction $(\alpha, \mathbf{b})$ is applied, the system moves from state $\mathbf{y}$ to state $\mathbf{y} + \mathbf{b}$. The corresponding rate is given by the propensity function. Therefore, we can rewrite the transition matrix of the CTMC as[3]:

$$q(\mathbf{y}, \mathbf{y}') = \begin{cases} \alpha(\mathbf{y}) & \text{if} \, \exists (\alpha, \mathbf{b}) \in \mathcal{R}, \mathbf{y}' = \mathbf{y} + \mathbf{b} \\ 0 & \text{otherwise} \end{cases}\,.$$

Next, we show that our counting abstractions have a natural interpretation as MPMs.

### 5.1 Node-Based Abstraction

First, we define the set of species $\mathcal{Z}$. Conceptually, species are node states which are aware of their partition:

$$\mathcal{Z} = \{(s, P) \mid s \in \mathcal{S}, P \in \mathcal{P}\}\,.$$

[3] Without loss of generality, we assume that different reactions have different change vectors. If this is not the case, we can merge reactions with the same update by summing their corresponding rate functions.

Again, we assume an implicit enumeration of $\mathcal{Z}$. We use $z.s$ and $z.P$ to denote the components of a give species $z$.

We can now represent the lumped CTMC state as a single population vector $\mathbf{y} \in \mathbb{Z}_{\geq 0}^{|\mathcal{Z}|}$, where $\mathbf{y}[z]$ the number of nodes belonging to species $z$ (i.e., which are in state $z.s$ and partition $z.P$). The image of the lumping function $\mathcal{L}$, i.e. the lumped state space $\mathcal{Y}$, is now a subset of non-negative integer vectors: $\mathcal{Y} \subset \mathbb{Z}_{\geq 0}^{|\mathcal{Z}|}$.

Next, we express the dynamics by a set of reactions. For each rule $r = s_1 \xrightarrow{f} s_2$ and each partition $P \in \mathcal{P}$, we define a reaction $(\alpha_{r,P}, \mathbf{b}_{r,P})$ with propensity function as:

$$\begin{aligned}\alpha_{r,P} :& \mathcal{Y} \to \mathbb{R}_{\geq 0}\\ \alpha_{r,P}(\mathbf{y}) =& \frac{1}{\mathcal{L}^{-1}(\mathbf{y})} \sum_{x \in \mathcal{L}^{-1}(\mathbf{y})} \sum_{n \in P} f(\mathbf{m}_{x,n}) \mathbb{1}_{x(n)=s_1} ,\end{aligned}$$

where $\mathbf{m}_{x,n}$ denotes the neighborhood vector of $n$ in network state $x$. Note that this is just the instantiation of Eq. 1 to the MPM framework.

The change vector $\mathbf{b}_{r,P} \in \mathbb{Z}^{|\mathcal{Z}|}$ is defined element-wise as:

$$\mathbf{b}_{r,P}[z] = \begin{cases} 1 & \text{if } z.s = s_2, P = z.P \\ -1 & \text{if } z.s = s_1, P = z.P \\ 0 & \text{otherwise} \end{cases} .$$

Note that $s_1, s_2$ refer to the current rule and $z.s$ to the entry of $\mathbf{b}_{r,P}$.

### 5.2 Edge-Based Counting Abstraction

We start by defining a *species neighborhood*. The species neighborhood of a node $n$ is a vector $\mathbf{v} \in \mathbb{Z}_{\geq 0}^{|\mathcal{Z}|}$, where $\mathbf{v}[z]$ denotes the number of neighbors of species $z$. We define $\mathcal{V}_n$ to be the set of possible species neighborhoods for a node $n$, given a fixed contact network and partitioning. Note that we still assume that a dummy node is used to encode the number of states in each partition.

Assuming an arbitrary ordering of pairs of states and partitions, we define

$$\begin{aligned}\mathcal{Z} = \big\{ (s_{source}, P_{source}, s_{target}, P_{target}) \, | s_{source}, s_{target} \in \mathcal{S}, P_{source}, P_{target} \in \mathcal{P}, \\ (s_{source}, P_{source}) \leq (s_{target}, P_{target}) \big\} .\end{aligned}$$

Let us define $\mathcal{V}_{\mathcal{P}}$ to be the set of partition neighborhoods all nodes in $P$ can have:

$$\mathcal{V}_P = \bigcup_{n \in P} \mathcal{V}_n .$$

For each rule $r = s_1 \xrightarrow{f} s_2$, and each partition $P \in \mathcal{P}$, and each $\mathbf{v} \in \mathcal{V}_P$, we define a propensity function $\alpha_{r,P,\mathbf{v}}$ with:

$$\begin{aligned}\alpha_{r,P,\mathbf{v}} :& \mathcal{Y} \to \mathbb{R}_{\geq 0}\\ \alpha_{r,P,\mathbf{v}}(\mathbf{y}) =& \frac{1}{\mathcal{L}^{-1}(\mathbf{y})} \sum_{x \in \mathcal{L}^{-1}(\mathbf{y})} \sum_{n \in P} f(\mathbf{m}_{x,n}) \mathbb{1}_{x(n)=s_1, V(n)=\mathbf{v}} .\end{aligned}$$

Note that the propensity does not actually depend on $\mathbf{v}$, it is simply individually defined for each $\mathbf{v}$. The reason for this is that the change vector depends on the a node's species neighborhood. To see this, consider a species $z = (s_{source}, P_{source}, s_{target}, P_{target})$, corresponding to edges connecting a node in state $s_{source}$ and partition $P_{source}$ to a node in state $s_{target}$ and partition $P_{target}$. There are two scenarios in which the corresponding counting variable has to change: (a) when the node changing state due to an application of rule $r$ is the source node, and (b) when it is the target node. Consider case (a); we need to know how many edges are connecting the updated node (which was in state $s_1$ and partition $P$) to a node in state $s_{target}$ and partition $P_{target}$. This information is stored in the vector $\mathbf{v}$, specifically in position $\mathbf{v}[s_{target}, P_{target}]$. The case in which the updated node is the target one is treated symmetrically. This gives rise to the following definition:

$$\mathbf{b}_{r,P,\mathbf{v}}[z] = \begin{cases} \mathbf{v}[z.s_{target}, z.P_{target}] & \text{if } s_2 = z.s_{source}, P = z.P_{source} \\ -\mathbf{v}[z.s_{target}, z.P_{target}] & \text{if } s_1 = z.s_{source}, P = z.P_{source} \\ \mathbf{v}[z.s_{source}, z.P_{source}] & \text{if } s_2 = z.s_{target}, P = z.P_{target} \\ -\mathbf{v}[z.s_{source}, z.P_{source}] & \text{if } s_1 = z.s_{target}, P = z.P_{target} \\ 0 & \text{otherwise} \end{cases}.$$

The first two lines of the definition handle cases in which the node changing state is the source node, while the following two lines deal with the case in which the node changing state appears as target.

Figure 4 illustrates how a lumped network state is influenced by the application of an infection rule.

### 5.3 Direct Construction of the MPM

Approximating the solution of an `SIS`-type process on a contact network by lumping the CTMC first, already reduces the computational costs by many orders of magnitude. However, this scheme is still only applicable when it is possible to construct the full CTMC in the first place. Recall that the number of network states is exponential in the number of nodes of the contact network, that is, $|\mathcal{X}| = |\mathcal{S}|^{|\mathcal{N}|}$.

However, in recent years, substantial effort was dedicated to the analysis of very small networks [25,32,35,37,48]. One reason is that when the size of a network increases, the (macro-scale) dynamics becomes more deterministic because stochastic effects tend to cancel out. For small contact networks, however, methods which capture the full stochastic dynamics of the system, and not only the mean behavior, are of particular importance.

A substantial advantage of the reduction to MPM is the possibility of constructing the lumped CTMC without building the full CTMC first. In particular, this can be done exactly for the node counting abstraction. On the other hand, for the edge counting we need to introduce an extra approximation in the definition of the rate function, roughly speaking introducing an approximate probability

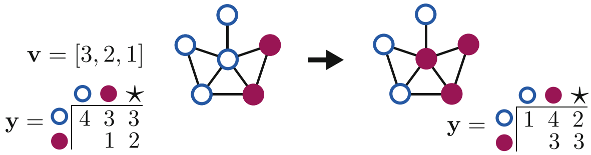


**Fig. 4.** Example of how the neighborhood $\mathbf{v}$ influences the update in the edge-based counting abstraction on an example graph. Here, all nodes belong to the same partition (thus, nodes states and species are conceptually the same) and the node states are ordered $[\mathtt{S}, \mathtt{I}, \star]$. The population vector $\mathbf{y}$ is given in matrix form for the ease of presentation.

distribution over neighboring vectors, as knowing how many nodes have a specific neighboring vector requires us full knowledge of the original CTMC. We present full details of such direct construction in the Appendix of [20].

### 5.4 Complexity of the MPM

The size of the lumped MPM is critical for our method, as it determines which solution techniques are computationally tractable and provides guidelines on how many partitions to choose. There are two notions of size to consider: (a) the number of population variables and (b) the number of states of the underlying CTMC. While the latter governs the applicability of numerical solutions for CTMCs, the former controls the complexity of a large number of approximate techniques for MPMs, like mean field or moment closure.

*Node-Based Abstraction.* In this abstraction, the population vector is of length $|\mathcal{S}| \cdot |\mathcal{P}|$, i.e. there is a variable for each node state and each partition.

Note that the sum of the population variables for each partition $P$ is $|P|$, the number of nodes in the partition. This allows us to count easily the number of states of the CTMC of the population model: for each partition, we need to subdivide $|P|$ different nodes into $|\mathcal{S}|$ different classes, which can be done in $\binom{|P|+|\mathcal{S}|-1}{|\mathcal{S}|-1}$ways, giving a number of CTMC states exponential in the number $|\mathcal{S}|$ of node states and $|\mathcal{P}|$ of partitions, but polynomial in the number of nodes:

$$|\mathcal{Y}| = \prod_{P \in \mathcal{P}} \binom{|P| + |\mathcal{S}| - 1}{|\mathcal{S}| - 1}.$$

*Edge-Based Abstraction.* The number of population variables, in this case, is one for each edge connecting two different partitions, plus those counting the number of nodes in each partition and each node state, due to the presence of the dummy state. In total, we have $\frac{q(q-1)}{2} + q$ population variables, with $q = |\mathcal{S}| \cdot |\mathcal{P}|$.

In order to count the number of states of the CTMC in this abstraction, we start by observing that the sum of all variables for a given pair of partitions $P', P''$ is the number of edges connecting such partitions in the graph. We use

$\epsilon(P', P'')$ to denote the number of edges between $P', P''$ (resp. the number of edges inside $P'$ if $P' = P''$). Thus,

$$|\mathcal{Y}| \leq \prod_{\substack{P', P'' \in \mathcal{P}^2 \\ P' \leq P''}} \binom{\epsilon(P', P'') + \mathcal{S}^2 - 1}{\mathcal{S}^2 - 1} \cdot \prod_{P \in \mathcal{P}} \binom{|P| + |\mathcal{S}| - 1}{|\mathcal{S}| - 1}.$$

This is an over-approximation, because not all combinations are consistent with the graph topology. For example, a high number of infected nodes in a partition might not be consistent with a small number of `I` − `I`-edges inside the partition. Note that also this upper bound is exponential in $|\mathcal{S}|$ and $|\mathcal{P}|$ but still polynomial in the number of nodes $N$, differently from the original network model, whose state space is exponential in $N$.

The exponential dependency on the number of species (i.e., dimensions of the population vector) makes the explicit construction of the lumped state space viable only for very small networks with a small number of node states. However, this is typically the case for spreading models like `SIS` or `SIR`. Yet, also the number of partitions has to be kept small, particularly in realistic models. We expect that the partitioning is especially useful for networks showing a small number of large-scale homogeneous structures, as happens in many real-world networks [12].

An alternative strategy for analysis is to derive mean-field [5] or moment closure equations [41] for MPMs, which can be done without explicitly constructing the lumped (and the original) state space. These are sets of ordinary differential equation (ODE) describing the evolution of (moments of) the population variables. We refer the reader to [10] for a similar approach regarding the node-based abstraction.

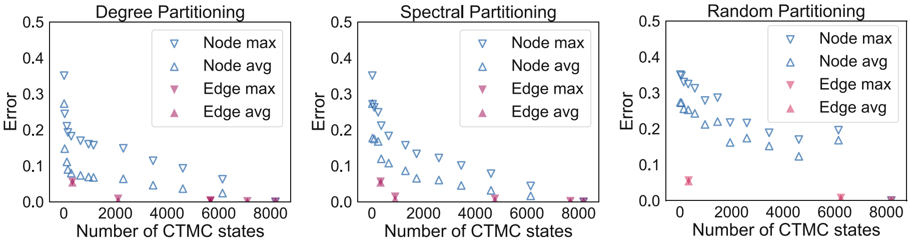


**Fig. 5.** Trade of between accuracy and state space size for the node-based (blue) and edge-based (magenta, filled) counting abstraction. Results are shown for node partitions based on the degree (l.), spectral embedding (c.), and random partitioning (r.). The accuracy is measured as the mean (△) and maximal (▽) difference between the original and lumped solution over all timepoints. (Color figure online)

## 6 Numerical Results

In this section, we compare the numerical solution of the original model—referred to as baseline model—with different lumped MPMs. The goal of this comparison is to provide evidence supporting the claim that the lumping preserves the dynamics of the original system, with an accuracy increasing with the resolution of the MPM. We will perform the comparison by solving numerically the ground and the lumped system, thus comparing the probability of each state in each point in time. In practical applications of our method, exact transient or steady state solutions may not be feasible, but in this case we can still rely to approximation methods for MPM [5, 41]. Determining which of those techniques performs best in this context is a direction of future exploration.

A limit of the comparison based on numerical solution of the CTMC is that the state space of the original model has $|\mathcal{S}|^{|\mathcal{N}|}$ states, which limits the size of the contact network strongly[4].

Let $P(X(t) = x)$ denote the probability that the baseline CTMC occupies network state $x \in \mathcal{X}$ at time $t \geq 0$. Furthermore, let $P(Y(t) = y)$ for $t \geq 0$ and $y \in \mathcal{Y}$ denote the same probability for a lumped MPM (corresponding to a specific partitioning and counting abstraction). To measure their difference, we first approximate the probability distribution of the original model using the lumped solution, invoking the lumping assumption which states that all network states which are lumped together have the same probability mass. We use $P_L$ to denote the *lifted* probability distribution over the original state space given a lumped solution. Formally,

$$P_L\big(Y(t) = x\big) = \frac{P\big(Y(t) = y\big)}{|\mathcal{L}^{-1}(y)|} \qquad \text{where } y \text{ is s.t. } L(x) = y.$$

We measure the difference between the baseline and a lumped solution at a specific time point by summing up the difference in probability mass of each state, then take the maximum error in time:

$$d(P, P_L) = \max_t \sum_{x \in \mathcal{X}} \Big| P_L\big(Y(t) = x\big) - P(X(t) = x) \Big| .$$

In our experiments, we used a small toy network with 13 nodes and 2 states ($2^{13} = 8192$ network states). We generated a synthetic contact network following the Erdős–Rényi graph model with a connection probability of 0.5. We use a `SIS` model with an infection rate of $\lambda = 1.0$ and a recovery rate of $\mu = 1.3$. Initially, we assign an equal amount of probability mass to all network states.

Figure 5 shows the relationship between the error of the lumped MPM, the type of counting abstraction and the method used for node partitioning. We also report the mean difference together with the maximal difference over time.

From our results, we conclude that the edge-based counting abstraction yields a significantly better trade-off between state space size and accuracy. However,

[4] Code is available at github.com/gerritgr/Reducing-Spreading-Processes.

it generates larger MPM models than the node-based abstraction when adding a new partition. We also find that spectral and degree-based partitioning yield similar results for the same number of CTMC states and that random partitioning performed noticeably worse, for both edge-based and node-based counting abstractions.

## 7 Conclusions and Future Work

This work developed first steps in a unification of the analysis of stochastic spreading processes on networks and Markov population models. Since the so obtained MPM can become very large in terms of species, it is important to be able to control the trade-off between state space size and accuracy.

However, there are still many open research problems ahead. Most evidently, it remains to be determined which of the many techniques developed for the analysis of MPMs (e.g. linear noise, moment closure) work best on our proposed epidemic-type MPMs and how they scale with increasing size of the contact network. We expect also that these reduction methods can provide a good starting point for deriving advanced mean-field equations, similar to ones in [10]. Moreover, literature is very rich in proposed moment-closure-based approximation techniques for MPMs, which can now be utilized [19,44]. We also plan to investigate the relationship between lumped mean-field equations [21,30] and coarse-grained counting abstractions further.

Future work can additionally explore counting abstraction of different types, for instance, a neighborhood-based abstraction like the one proposed by Gleeson in [13,14].

Finally, we expect that there are many more possibilities of partitioning the contact network that remain to be investigated and which might have a significant impact on the final accuracy of the abstraction.

**Acknowledgements.** This research has been partially funded by the German Research Council (DFG) as part of the Collaborative Research Center "Methods and Tools for Understanding and Controlling Privacy". We thank Verena Wolf for helpful discussions and provision of expertise.

## References

1. Allen, G.E., Dytham, C.: An efficient method for stochastic simulation of biological populations in continuous time. Biosystems **98**(1), 37–42 (2009)
2. Barabási, A.-L.: Network Science. Cambridge University Press, Cambridge (2016)
3. Bobbio, A., Cerotti, D., Gribaudo, M., Iacono, M., Manini, D.: Markovian agent models: a dynamic population of interdependent Markovian agents. In: Al-Begain, K., Bargiela, A. (eds.) Seminal Contributions to Modelling and Simulation. SFMA, pp. 185–203. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-33786-9_13
4. Bortolussi, L.: Hybrid behaviour of Markov population models. Inf. Comput. **247**, 37–86 (2016)

5. Bortolussi, L., Hillston, J., Latella, D., Massink, M.: Continuous approximation of collective system behaviour: a tutorial. Perform. Eval. **70**(5), 317–349 (2013)
6. Buchholz, P.: Exact and ordinary lumpability in finite Markov chains. J. Appl. Probab. **31**(1), 59–75 (1994)
7. Cao, Y., Gillespie, D.T., Petzold, L.R.: Efficient step size selection for the tau-leaping simulation method. J. Chem. Phys. **124**(4) (2006)
8. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: ERODE: a tool for the evaluation and reduction of ordinary differential equations. In: Legay, A., Margaria, T. (eds.) TACAS 2017. LNCS, vol. 10206, pp. 310–328. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54580-5_19
9. Cota, W., Ferreira, S.C.: Optimized gillespie algorithms for the simulation of markovian epidemic processes on large and heterogeneous networks. Comput. Phys. Commun. **219**, 303–312 (2017)
10. Devriendt, K., Van Mieghem, P.: Unified mean-field framework for susceptible-infected-susceptible epidemics on networks, based on graph partitioning and the isoperimetric inequality. Phys. Rev. E **96**(5), 052314 (2017)
11. Gan, C., Yang, X., Liu, W., Zhu, Q., Zhang, X.: Propagation of computer virus under human intervention: a dynamicalmodel. Discrete Dyn. Nature Soc. **2012**, 8 (2012)
12. Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. **99**(12), 7821–7826 (2002)
13. Gleeson, J.P.: High-accuracy approximation of binary-state dynamics on networks. Phys. Rev. Lett. **107**(6), 068701 (2011)
14. Gleeson, J.P.: Binary-state dynamics on complex networks: pair approximation and beyond. Phys. Rev. X **3**(2), 021004 (2013)
15. Goltsev, A., De Abreu, F., Dorogovtsev, S., Mendes, J.: Stochastic cellular automata model of neural networks. Phys. Rev. E **81**(6), 061921 (2010)
16. Goutsias, J., Jenkinson, G.: Markovian dynamics on complex reaction networks. Phys. Rep. **529**(2), 199–264 (2013)
17. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: a survey. Knowl.-Based Syst. **151**, 78–94 (2018)
18. Grima, R.: An effective rate equation approach to reaction kinetics in small volumes: theory and application to biochemical reactions in nonequilibrium steady-state conditions. J. Chem. Phys. **133**(3), 035101 (2010)
19. Grima, R.: A study of the accuracy of moment-closure approximations for stochastic chemical kinetics. J. Chem. Phys. **136**(15), 04B616 (2012)
20. Großmann, G., Bortolussi, L.: Reducing spreading processes on networks to Markov population models. arXiv preprint arXiv:1906.11508 (2019)
21. Großmann, G., Kyriakopoulos, C., Bortolussi, L., Wolf, V.: Lumping the approximate master equation for multistate processes on complex networks. In: McIver, A., Horvath, A. (eds.) QEST 2018. LNCS, vol. 11024, pp. 157–172. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99154-2_10
22. Großmann, G., Wolf, V.: Rejection-based simulation of stochastic spreading processes on complex networks. arXiv preprint arXiv:1812.10845 (2018)
23. Hagberg, A., Swart, P., Chult, D.S.: Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab. (LANL), Los Alamos, NM, United States (2008)
24. 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

25. Holme, P.: Shadows of the susceptible-infectious-susceptible immortality transition in small networks. Phys. Rev. E **92**(1), 012804 (2015)
26. Keeling, M.J., Rohani, P.: Modeling Infectious Diseases in Humans and Animals. Princeton University Press, Princeton (2011)
27. KhudaBukhsh, W.R., Auddy, A., Disser, Y., Koeppl, H.: Approximate lumpability for Markovian agent-based models using local symmetries. arXiv:1804.00910
28. Kiss, I.Z., Miller, J.C., Simon, P.L.: Mathematics of epidemics on networks: from exact to approximate models. Forthcoming in Springer TAM series (2016)
29. Kitsak, M., et al.: Identification of influential spreaders in complex networks. Nat. Phys. **6**(11), 888 (2010)
30. Kyriakopoulos, C., Grossmann, G., Wolf, V., Bortolussi, L.: Lumping of degree-based mean-field and pair-approximation equations for multistate contact processes. Phys. Rev. E **97**(1), 012301 (2018)
31. Li, G., Rabitz, H.: A general analysis of approximate lumping in chemical kinetics. Chem. Eng. Sci. **45**(4), 977–1002 (1990)
32. López-García, M.: Stochastic descriptors in an sir epidemic model for heterogeneous individuals in small networks. Math. Biosci. **271**, 42–61 (2016)
33. Mateescu, M., Wolf, V., Didier, F., Henzinger, T.: Fast adaptive uniformisation of the chemical master equation. IET Syst. Biol. **4**(6), 441–452 (2010)
34. May, R.M., Arinaminpathy, N.: Systemic risk: the dynamics of model banking systems. J. R. Soc. Interface **7**(46), 823–838 (2009)
35. Moslonka-Lefebvre, M., Pautasso, M., Jeger, M.J.: Disease spread in small-size directed networks: epidemic threshold, correlation between links to and from nodes, and clustering. J. Theor. Biol. **260**(3), 402–411 (2009)
36. Ng, T.W., Turinici, G., Danchin, A.: A double epidemic model for the sars propagation. BMC Infect. Dis. **3**(1), 19 (2003)
37. Pautasso, M., Moslonka-Lefebvre, M., Jeger, M.J.: The number of links to and from the starting node as a predictor of epidemic size in small-size directed networks. Ecol. Complex. **7**(4), 424–432 (2010)
38. Porter, M., Gleeson, J.: Dynamical Systems on Networks: A Tutorial, vol. 4. Springer, Switzerland (2016). https://doi.org/10.1007/978-3-319-26641-1
39. Rodrigues, H.S.: Application of sir epidemiological model: new trends. arXiv:1611.02565 (2016)
40. Rodrigues, H.S., Monteiro, M.T.T., Torres, D.F.: Dynamics of dengue epidemics when using optimal control. Math. Comput. Modell. **52**(9–10), 1667–1673 (2010)
41. Schnoerr, D., Sanguinetti, G., Grima, R.: Approximation and inference methods for stochastic biochemical kinetics - a tutorial review. J. Phys. A **51**, 169501 (2018)
42. Simon, P.L., Taylor, M., Kiss, I.Z.: Exact epidemic models on graphs using graph-automorphism driven lumping. J. Math. Biol. **62**(4), 479–508 (2011)
43. Singh, A., Hespanha, J.P.: Stochastic hybrid systems for studying biochemical processes. Roy. Soc. A **368**(1930), 4995–5011 (2010)
44. Soltani, M., Vargas-Garcia, C.A., Singh, A.: Conditional moment closure schemes for studying stochastic dynamics of genetic circuits. IEEE Trans. Biomed. Circuits Syst. **9**(4), 518–526 (2015)
45. St-Onge, G., Young, J.-G., Hébert-Dufresne, L., Dubé, L.J.: Efficient sampling of spreading processes on complex networks using acomposition and rejection algorithm. Comput. Phys. Commun. **240**, 30–37 (2019)
46. Van Kampen, N.G.: Stochastic Processes in Physics and Chemistry, vol. 1. Elsevier, Amsterdam (1992)
47. Van Mieghem, P., Omic, J., Kooij, R.: Virus spread in networks. IEEE/ACM Trans. Networking **17**(1), 1–14 (2009)

48. Ward, J.A., Evans, J.: A general model of dynamics on networks with graph automorphism lumping. In: Aiello, L.M., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L.M. (eds.) COMPLEX NETWORKS 2018. SCI, vol. 812, pp. 445–456. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05411-3_36
49. Watts, D.J., Dodds, P.S.: Influentials, networks, and public opinion formation. J. Consum. Res. **34**(4), 441–458 (2007)
50. Wei, J., Kuo, J.C.: Lumping analysis in monomolecular reaction systems: analysis of the exactly lumpable system. Ind. Eng. Chem. Fundam. **8**(1), 114–123 (1969)
51. Wei, X., Valler, N.C., Prakash, B.A., Neamtiu, I., Faloutsos, M., Faloutsos, C.: Competing memes propagation on networks: a network science perspective. IEEE J. Sel. Areas Commun. **31**(6), 1049–1060 (2013)
52. Zhao, L., Cui, H., Qiu, X., Wang, X., Wang, J.: Sir rumor spreading model in the new media age. Phys. A **392**(4), 995–1003 (2013)
53. Zhao, L., Wang, J., Chen, Y., Wang, Q., Cheng, J., Cui, H.: Sihr rumor spreading model in social networks. Phys. A **391**(7), 2444–2453 (2012)