Lumping Reductions for Multispread in Multi-Layer Networks

Tatjana Petrov$^{1,2[0000-0002-9041-0905]}$ and Stefano Tognazzi$^{1,2[0000-0002-3557-2426]}$

[1] Universität Konstanz, Konstanz, Germany
[2] Centre for the Advanced Study of Collective Behaviour, Konstanz, Germany.
{tatjana.petrov,stefano.tognazzi}@uni-konstanz.de

Abstract. Spreading phenomena such as epidemics emerge from simple, local interactions among a large number of agents, influencing each-other through different networks of interaction. Computational modelling and analysis of spreading processes quickly becomes challenging, due to the combinatorial explosion of possible network configurations, evolving stochastically over time, resulting into a large-scale continuous-time Markov chain (CTMC). Unlike single-layer networks, multi-layer networks (MLNs) allow to simultaneously incorporate multiple layers of relationship between network nodes, as well as inter-layer correlations, in a natural and compact way. However, more detailed description further challenges the respective computational simulation and analysis.

In this paper, we first propose a number of formal model reduction techniques for MLNs. The techniques are inspired by the state lumping ideas previously used in context of reductions of CTMCs (preserving dynamical features), and reductions of static, undirected networks (preserving structural properties such as network centrality). We then experimentally show and compare the performance of different reductions over a variety of artificially generated and real-world MLNs. Finally, we show how to efficiently compute the proposed reductions, and we show how to speed up the respective computational simulation of complex spreading processes on MLNs.

Keywords: Multi-Layer Networks · Spreading Processes · Model Reduction Techniques · Lumping · Stochastic Processes

1 Introduction

Spreading phenomena such as epidemics emerge from simple, local interactions among a large number of agents, influencing each-other through different networks of interaction. The ability to faithfully model and predict the macroscopic consequences of spreading phenomena over networks, is of key importance in a wide range of application scenarios, ranging from mitigating epidemics [22], to understanding animal collectives [9], online social networks [30], all the way to the design and verification of self-organised cyber-physical systems such as the internet-of-things or a network of robots [14].

Computational modelling and analysis of spreading processes quickly becomes challenging, due to the combinatorial explosion of possible network configurations, typically evolving stochastically over time, resulting into a large-scale continuous-time Markov chain (CTMC) [29, 12, 4]. For instance, a network of n agents where each node can either be infected (I) or susceptible (S) to infection, gives rise to 2^n possible network configurations. Reaction network formalism facilitates the description of agents interacting based on their feature described by a suitable variable name. For example, a Susceptible-Infected-Susceptible (SIS) model, widely used to study the spread of opinions, rumours and memes in social networks, is specified through two local interactions between agents in state S or I: (i) $S + I \rightarrow 2I$ for infection spread and (ii) $I \rightarrow S$ for recovery. Such model typically does not specify whether two agents are related or not and hence assumes population homogeneity. Homogeneity assumption allows to reduce the number of states through a *population abstraction*, requiring to only enumerate the total number of agents in each of the states (S and I), hence reducing the number of states in the above example from 2^n to n.

Network homogeneity is however a strong assumption in most real-world scenarios. For instance, in case of epidemic spread, different individuals will have a different range and intensity of interactions with family, friends and coworkers that form a network of physical contact in which an infection can spread. State representation in form of a single-layered network configuration (where each node has a specific state, e.g. S or I) allows to encode such relational heterogeneity. Reaction network formalism can facilitate such encoding through identifiers denoting position of nodes in a network, e.g. $S_i + I_j \rightarrow I_i + I_j$, and conditioning the respective rate to the existence of an edge between nodes [12, 4, 11]. Moreover, in reality, the same group of individuals can partake in different spreading processes at different interaction networks, which in turn affect each-other [2]. In these cases, independent analysis of spreading processes over single-layer networks is limited. To exemplify, while the infection spreads through physical contact, the rate of contact is influenced by other factors, such as awareness: an agent aware of the disease will have less contact; In turn, an agent with the disease will spread awareness more actively. Awareness will spread as well, through a communication network that may have significantly different dynamics than the infection spread. A faithful representation of interrelated spreading processes will further blow-up the space of states and parameters in a model [11].

Unlike single-layer networks, representing system state in terms of a multi-layer network (MLNs) allows to simultaneously incorporate multiple layers of relationship between network nodes, as well as inter-layer correlations, in a natural and compact way [11, 23, 7, 16] . However, more detailed description further challenges the respective computational simulation and analysis. Formal reductions based on lumping states that are behaviourally equivalent are desirable [26], yet novel techniques are needed for the context of MLNs.

In this paper, we propose a number of formal model reduction techniques for MLNs. The techniques are inspired by reductions detecting symmetries that aim to provably preserve the properties of the original system, and are effectuated algorithmically through lumping states. Different state representations and respective semantics of executions are subject to different lumping techniques, including the state lumping ideas previously used in context of reductions of CTMCs and differential-drift dynamical systems (preserving dynamical features), as well as reductions of static, undirected networks (preserving structural properties such as network centrality). We then empirically demonstrate and compare the performance of different reductions over a variety of artificially generated and real-world MLNs. Finally, we show how to efficiently compute the proposed reductions, and we show how to speed up the respective computational simulation of complex spreading processes on MLNs.

Related works. Even under the assumption of homogeneity, reaction networks, or Petri nets, often become computationally expensive to analyse, especially in context of systems and synthetic biology, when modelling gene regulation and signal transduction, the number of possible agent states (chemical species) is combinatorial, and respective multiplicities vary in scale. To this end, a large body of reduction techniques have been proposed. One popular approach are mean-field approximations, such as the deterministic limit, where system state has the dimension equal to the agent states (e.g. two states in case of SIS model), and evolves deterministically according to a system of differential equations [18]. However, the deterministic approximation is satisfactory only for special system instances, and does not faithfully describe the mean-field dynamics in case of complex contagion and low agent counts. Other existing works are based on exploiting symmetries and/or multi-scaleness of reaction networks, complemented with powerful tool support [4].

2 Background

Notation. Throughout this work, when clear from context, we will use x_i both to denote the i-th element of vector $\boldsymbol{x}$ or the value of the map $x(i)$. For a partition $\mathcal{H}$ over a variable set $V_p \subseteq \{\mathrm{x}_1, \mathrm{x}_2, \ldots\}$, induced by an equivalence relation $\sim_{\mathcal{H}} \subseteq V_p \times V_p$, we will denote elements of a partition class $H \in \mathcal{H}$ by $\mathrm{x}_{H,1}, \mathrm{x}_{H,2}, \ldots, \mathrm{x}_{H,|H|}$. We denote by $\|\cdot\|_1$ the 1-norm. We will denote with $V_N = \{1, \ldots, N\}$, $V_L = \{1, \ldots, L\}$ the set of nodes and layers, respectively. We will assume $N > 0$ and $L > 1$ (MLNs with $L = 1$ are a special case that corresponds with traditional single-layer networks). Vectors will be assumed to be written in column notation.

2.1 Reaction Networks

A reaction network is formally a pair (S, R) where S is a set of species and R is a set of reactions. Each reaction is in the form $\rho \xrightarrow{\alpha} \pi$, where $\alpha > 0$ is a kinetic parameter and ρ and π are multisets of species called reactants and products, respectively. The multiplicity of species S in ρ is denoted with $\rho(S)$, which represents the stochiometry coefficient. The set of all reagents and products across all the reactions in the network are denoted by $\rho(R)$ and $\pi(R)$. Throughout this work, we will consider two different semantics for reaction networks: the first one is a *stochastic* semantics while the second one is a *deterministic* semantics.

Stochastic semantics. The stochastic semantics of a reaction network is given by a Continuous Time Markov Chain (CTMC) where each state σ is a multiset of species. From a state σ such that $\rho \subset \sigma$, a reaction $\rho \xrightarrow{\alpha} \pi$ induces a transition with mass-action propensity $\alpha \prod_{S \in \rho} \binom{\sigma(S)}{\rho(S)}$ to state $\sigma + \pi - \rho$, where the plus and minus operators indicate multiset union and difference, respectively, while $S \in \rho$ denotes that S belongs to the support of $\rho(S)$, i.e. $\rho(S) > 0$. Given an initial state $\hat{\sigma}$, the state space can be derived by exhaustively applying the reactions to compute all possible states reachable from $\hat{\sigma}$. We denote $out(\sigma)$ the multiset of outgoing transitions from state σ,

$$out(\sigma) = \{\sigma \xrightarrow{\lambda} \sigma + \pi - \rho \mid (\rho \xrightarrow{\alpha} \pi) \in \mathcal{R}, \lambda = \alpha \prod_{S \in \rho} \binom{\sigma(S)}{\rho(S)}\} \tag{1}$$

For any two distinct states σ and ϕ, we denote by $q(\sigma, \phi)$ the sum of the propensities from σ to ϕ across all the reactions, that is

$$q(\sigma, \phi) = \sum_{(\sigma \xrightarrow{\lambda} \phi) \in out(\sigma)} \lambda \tag{2}$$

Moreover, we set $q(\sigma, \sigma)$ to be the negative sum of all possible transitions from state σ, i.e., $q(\sigma, \sigma) = -\sum_{\phi \neq \sigma} q(\sigma, \phi)$. These values ensure a well-formed CTMC generator matrix, which characterises the dynamical evolution of the CTMC. Each component of its solution, is the probability of being in a given multiset of species at time t starting from some initial probability distribution.

Deterministic semantics The deterministic semantics of a reaction network (S, R) is a system of Ordinary Differential Equations (ODEs) in which each variable corresponds to a species of the reaction network. $|S| = s$, $|R| = r$, $x_j \in S$:

$$\dot{x}_j = \sum_{i=1}^{r} (\pi(x_j) - \rho(x_j)) \alpha_i \prod_{x_k \in S} x_k^{\rho(S_k)} \tag{3}$$

for $1 < j < s$ and S_k indicating the species corresponding to variable x_k. The deterministic model is a limit of the stochastic model when all species in a reaction network are highly abundant [18].

2.2 Spreading (epidemic) processes on Single-Layer Networks

Let $G = (V, E)$ be a graph (single-layer network) where V is the set of nodes and E a list of un-directed edges. Let $A = (a_{ij})$ be the adjacency matrix of G with $A \in \mathbb{R}^{|V| \times |V|}$. A SIS epidemic process can be described by the reaction network [12, 22, 26, 4]

$$S_i + I_j \xrightarrow{a_{ij}\beta} I_i + I_j$$

$$I_i \xrightarrow{\mu} S_i$$

with $1 \leq i, j \leq |V|$, $j \neq i$. Where the first reaction models infections by neighbors and the second reaction is the spontaneous recovery, with parameters β and μ respectively. A physically meaningful initial state σ must be such that every node i is initially infected ($\hat{\sigma}_{S_i} = 0$, $\hat{\sigma}_{I_i} = 1$) or susceptible ($\hat{\sigma}_{S_i} = 1$, $\hat{\sigma}_{I_i} = 0$).

2.3 Multiplex Multi-Layer Networks

In this paper, we will work with a generalisation of networks called *multiplex networks* or *edge-colored-graphs*, which are useful for simultaneously representing different kinds of relationships over the same set of nodes [7]. This paper will focus on undirected multiplex networks.

Definition 1. *A* multiplex network *with N nodes and L layers is an ordered collection of L undirected graphs over the same set of nodes:*

$$\mathcal{G} = \{G^{(l)} = (V_N, E^{(l)})\}_{l \in V_L},$$

where $E^{(l)} : V_N \times V_N \to \mathbb{R}_{\geq 0}$ are the edges on layer $l \in V_L$. For every layer l, we denote the non-negative adjacency matrix of the graph $G^{(l)}$ by $\mathbf{A}^{(l)} = (A^{(l)}_{ij}) \in \mathbb{R}^{N \times N}_{\geq 0}$. Then, the multiplex network can be represented by a 3^{rd}-order adjacency tensor*:*

$$\mathcal{A} = (\mathcal{A}_{ijl}) \in \mathbb{R}^{N \times N \times L}_{\geq 0}, \textit{ such that } \mathcal{A}_{ijl} := A^{(l)}_{ij} = E^{(l)}(i,j),$$

that is, $\mathcal{A}_{ijl}$ represents the presence of an edge between nodes i and j on layer l. Throughout this work we will consider unweighted networks unless stated otherwise.

2.4 Interacting spreading processes on Multiplex Multi-Layer Networks

While SIS models have been used to study the dynamics of epidemic processes it is often the case that it is not enough to characterise all the facets of the complex interaction among agents. Variants of SIS models, such as SIR, SIER have been proposed but they mainly account for one diffusion process [22]. In real-world scenarios it is often the case that multiple spreading processes act simultaneously on the network. For instance, different infections can spread at the same time on the same contact network with different infection rates [25]. Competing opinions can form on the same online (or real-life) social network [30] or, as in the case we will mainly focus on in this work, competing spreading processes can be dependent on each other while spreading on different networks [11].

In this work, we will mainly focus on a UAU-SIS process proposed in [11]. We use a multiplex with two layers: one layer represents the *physical* persistent social contacts (nodes that can infect each other), while the second layer represents the *virtual* contacts (nodes that communicate with each other, not necessarily in physical contact, e.g. online social network peers, etc.). In the physical layer a node can be either in state *susceptible* (S) or *infected* (I). The infection propagates from an infected node to its neighbors with rate β and infected nodes recover and become susceptible again with rate μ. Simultaneously, on the virtual layer an UAU process takes place, nodes spread awareness about the epidemics. The states of this process are *unaware* (U) or *aware* (A) of the existence of the epidemic and its prevention. Unaware individuals do not have any information on how to prevent infection, while aware individuals reduce their risk of to be infected. In this model, individuals can become aware in two different ways: they can be informed from aware neighbors (with rate λ) or, it is assumed that an infected individual automatically becomes aware of the infection. Moreover, it is commonly assumed, that, since the awareness is bound with the seasonality of an epidemic, an aware individual can become unaware with rate δ. In this scenario, we will consider two different infection rate β: if a node is susceptible and unaware, it will be infected with rate β^U; On the other hand, if a node is susceptible and aware the rate of which he will be infected is reduced by a factor of γ and its infection rate will be $\beta^A = \gamma\beta^U$, where complete immunity can be modeled by fixing $\gamma = 0$.

Under the above assumptions, every node can be in one of the following three states: unaware and susceptible (US), aware and susceptible (AS) or aware and infected (AI). Note that the state unaware and infected (UI) is spurious under the assumption that a node automatically becomes aware if infected. Let us consider the multiplex network $\mathcal{G}$ with two layers, we denote with $A^{(1)} = (a_{ij})$ the adjacency matrix of the first (physical contact) layer and we denote with $A^{(2)} = B = (b_{ij})$ the adjacency matrix of the second (virtual interaction) layer. We can now write the process in terms of the following reaction network:

$$AS_i \xrightarrow{\delta} US_i$$

$$AI_i \xrightarrow{\mu} AS_i$$

$$AS_i + AI_j \xrightarrow{a_{ij}\beta^A} AI_i + AI_j$$

$$US_i + AI_j \xrightarrow{a_{ij}\beta^U} AI_i + AI_j$$

$$US_i + AI_j \xrightarrow{b_{ij}\lambda} AS_i + AI_j$$

$$US_i + AS_j \xrightarrow{b_{ij}\lambda} AS_i + AS_j$$

Throughout this work we consider this spread process as the new application of a variety of lumping techniques.

2.5 Lumping states in a CTMC

We first introduce lumping reductions at the level of Markov chain. Let us consider a CTMC over the state spaces $\{\sigma_1, \ldots, \sigma_n\}$ and let Q be its generator matrix where $Q(i,j) \geq 0$ denotes the rate at which the CTMC moves from state i to state j. Lumpability notions, first introduced in [15] and generalised in [3], allow to define a CTMC over a smaller, aggregate state space, which faithfully abstracts the original one.

Let $\mathcal{H}$ be a partition of $\{\sigma_1, \ldots, \sigma_n\}$.

- CTMC is *ordinarily lumpable* with respect to partition $\mathcal{H}$ if any two states σ_1, σ_2 lumped together in a partition block have equal aggregate rates towards any partition block $H' \in \mathcal{H}$, that is, $\forall H, H'. \forall \sigma_1, \sigma_2 \in H. \sum_{\sigma \in H'} q(\sigma, \sigma_1) = \sum_{\sigma \in H'} q(\sigma, \sigma_2)$. Ordinary lumpability guarantees that the aggregate CTMC process is equivalent in distribution to the projection of the original CTMC process to the partition blocks, independently of the initial distribution [10].

- CTMC is *exactly lumpable* with respect to partition $\mathcal{H}$ if any two states σ_1, σ_2 lumped together in a partition block have equal aggregate rates stemming from any partition block $H \in \mathcal{H}$, that is, $\forall H, H'. \forall \sigma_1, \sigma_2 \in H'. \sum_{\sigma \in H} q(\sigma, \sigma_1) = \sum_{\sigma \in H} q(\sigma, \sigma_2)$.
  Exact lumpability guarantees that the aggregate CTMC process is equivalent in distribution to the projection of the original CTMC process to the partition blocks, in the case when initial distributions among the aggregate states are uniform [10]. The exact lumpability criterion ensures that the uniform distribution over the initial distribution remains invariant throughout the system evolution. Therefore, exact lumpability provides a stronger guarantee about the aggregate CTMC: the transient distribution of the original process can be faithfully reproduced from the lumped one at any point of time before as well as at the stationary distribution [10].

Given an ordinary (resp. exactly) lumpable partition, a lumped CTMC can be constructed by associating an aggregate, macro-state to each block; transitions between macro-states are labelled with the aggregate outgoing (resp. incoming) rate.

An ordinary lumpable partition is in general not necessarily exactly lumpable and vice-versa, while there exist partitions that are both exactly and ordinarily lumpable [8]. It is worth mentioning that ordinary and/or exactly lumpable partitions are not the maximal ones, as well as that more general notions of exact lumpability exist, where the invariant distribution among lumped states is non-uniform [10, 8];

Verifying the conditions for exact or ordinary lumpability over the full enumeration of the CTMC state space is impractical, as the state space which grows combinatorially with the multiplicities of initial state and the number of reactions. In order to bypass the enumeration of the CTMC state space, and yet guarantee that lumpability relations will hold, it is instructive to perform partitioning at the level of symbolic description of the local rules that generate the stochastic process at hand. For instance, given a list of reactions over a set of species, the lumping can be performed at the level of species. Then, lumping at the level of species will naturally lift to a lumping at the level of species' multi-sets, which can be inductively defined as follows: for all species $s, s' \in S$ and all multi-sets $\sigma, \sigma' \in \Sigma$, if $s \sim_S s'$ and $\sigma \sim_C \sigma'$, it will hold that $\sigma + s \sim_C \sigma' + s'$.

2.6 Lumping species in a reaction network

We next review three formal reductions techniques lumping species, based on the reaction network description [5]. Each of the techniques was proposed with a goal to guarantee a certain semantic relationships. These reduction ideas will be employed for reducing spreading processes over MLNs.

Let (S, R) be the reaction network. Then,

- (Forward Equivalence) $\sim_{FE} \subseteq S \times S$ is a forward equivalence, if sum of the drift functions in the respective differential semantics for any two equivalent states is equivalent (up to $\sim_{FE}$). The condition guarantees that the sum of solutions for species lumped by $\sim_{FE}$ will be equal to the solution of respective macro-species in the reduced ODE system.

Given a reaction network, finding relation $\sim_{FE}$ can be done in polynomial time (with Paige-Tarjan algorithm) [5]. It is worth mentioning that $\sim_{FE}$ refers to the differential semantics of reaction networks with mass-action kinetics, where the drifts of species' concentration variables are polynomial functions. A more general notion of $\sim_{FE}$ is termed in literature as *forward differential equivalence* (FDE), which we will denote by $\sim_{FDE}$. FDE refers to general systems of coupled differential equations, hence encompassing a wider class of differential semantics of reaction networks, including e.g. Hill kinetics or threshold drifts. Finding an FDE for a general system of differential equations however is coNP-hard [5].

- (Backward Equivalence) On the other hand, $\sim_{BE} \subseteq S \times S$ if the drift functions in the respective differential semantics for any two equivalent states are equivalent (up to $\sim_{BE}$). The lumping condition guarantees that, in case two lumped species start from the same initial conditions, their solutions in the deterministic semantics will coincide across time.
  The complexity of finding $\sim_{BE}$ is the same as for $\sim_{FE}$. Also, similarly, backward differential equivalence, denoted by $\sim_{BDE}$, refers to a more general notion of backward equivalence, applicable to any system of coupled differential equations, however at the price of increasing the computational complexity.
- (Stochastic Equivalence) $\sim_{SE} \subseteq S \times S$ is a stochastic equivalence, where two species are lumped, if the cumulative rates towards any partition of multi-sets inherited by $\sim_{SE}$ from any multi-set containing s and resp. s' are equal. This cumulative rate will represent the rate between the respective partitions of multi-sets in the reduced model.
  The condition guarantees that the partition over the CTMC states inherited from the partiting over the species set ($\sim_{SE}$) will be ordinary lumpable [4]. Finding the partition $\sim_{SE}$ is polynomial in the size of reaction network $O(|R||S|log|S|)$ (c.f. that finding a BE partition is $O(|R|log|S|)$). This significantly improves the complexity of searching for the lumpable partition directly over the expanded CTMC.

It is worth noticing that Kolmogorov forward equations for the CTMC are a special case of a polynomial system of ODEs, where variables represent probabilities of being in any of the reachable multi-sets. Applying the FE (resp. BE) to this system of equations induces an ordinary (resp. exactly) lumpable partition, but is identified at complexity polynomial in the number of reachable multi-sets, which is prohibitevely large in practice. Since in BDE every variable in the same equivalence class has the same solution, the original model can be fully recovered. On the other hand, from the quotient FDE model one cannot recover the original solutions, but FDE poses no restriction on the initial conditions.

2.7 Lumping nodes in multiplex networks

Lumping can also facilitate more efficient algorithms for computing structural properties of nodes within a network. In recent works [23], we use formal reductions to compute one notion of eigenvector centrality for multiplex MLNs, proposed in [28]. The centrality is defined through a 2-map, ***f****-eigenvector centrality*, in which the first component of the map represents the centrality associated to the *nodes*, while the second component is centrality associated to the *layers*.

Definition 2. *([28])* Let $\boldsymbol{\mathcal{A}} \in \mathbb{R}_{\geq 0}^{N \times N \times L}$ be the adjacency tensor of an MLN with weighted, undirected layers, and let $\alpha, \beta > 0$ be such that $\frac{2}{\beta} < (\alpha - 1)$. Then, define $\boldsymbol{f} = (f_1, f_2) : \mathbb{R}_{\geq 0}^{N} \times \mathbb{R}_{\geq 0}^{L} \to \mathbb{R}_{\geq 0}^{N} \times \mathbb{R}_{\geq 0}^{L}$ as follows:

$$f_1(\boldsymbol{x}, \boldsymbol{t})_i = \left(\sum_{j=1}^{N} \sum_{l=1}^{L} A_{ijl} x_j t_l \right)^{\frac{1}{\alpha}} \text{ for } i \in V_N,\; f_2(\boldsymbol{x}, \boldsymbol{t})_l = \left(\sum_{i=1}^{N} \sum_{j=1}^{N} A_{ijl} x_i x_j \right)^{\frac{1}{\beta}} \text{ for } l \in V_L.$$

In words, the centrality x_i of node i is a sum of the centralities of each of its neighbouring nodes, weighted by the product of the edge-weight and the centrality of the layer at which that connection lies. At the same time, the centrality of a layer t_l is a sum of the centrality of all edges at that layer, where an importance of an edge is, in addition to its own weight, weighted by the centrality of the two nodes which constitute it. The parameters α and β are introduced in order to guarantee convergence and respectively well-definedness in case of undirected MLNs. [3] In [28] the centrality vector of the nodes and layers is denoted by $(\boldsymbol{x}^*, \boldsymbol{t}^*) \in \mathbb{R}_{\geq 0}^{N} \times \mathbb{R}_{\geq 0}^{L}$ which is a limit of an iterative scheme based on the 2-map f.

- (Node Equivalence) $\sim_{NE} \subseteq V_N \times V_N$ is a node equivalence, where two nodes are lumped, if they have the same $\boldsymbol{f}$-eigenvector centrality value (i.e., $x_i \sim_{NE} x_j$, if $x_i^* = x_j^*$). Finding the partition $\sim_{NE}$ is polynomial in the size of the MLN $O(|E| log(|V_N| + |V_L|))$ where $|E|$ denotes the total number of edges in all the layers [23].

3 Results

In this work, we propose and evaluate the performance of the lumping techniques reviewed in Section 2 to multi-layer networks (MLNs). Specifically, we focus on the UAU-SIS process presented in Section 2.4 on MLNs. As a default, we will use $\delta = 0.6$, $\mu = 0.4$, $\lambda = 0.15$, $\beta^U = 0.4$ and $\beta^A = 0.01$. We will use two different sets of benchmarks: real-world networks from the Koblenz Network Collection [17], and a set of synthetic networks. The synthetic networks will have a physical layer built using a power-law degree distribution network generated with a configuration model with exponent 2.5. In all case studies, the virtual layer is a copy of the physical layer network with an added percentage of random edges (non-overlapping with previous edges).

We showcase our findings with the aim of obtaining the maximal aggregation, if not specified differently. The aim of the *maximal aggregation* approach is to obtain the smallest possible reduced system. As the techniques presented in this paper rely on partition refinement procedures, in order to find the smallest possible reduction, the goal is to find the coarsest partition refinement of the initial partition. For this reason, we start with the trivial initial partition in which all the nodes belong to the same block. For SIS or similar models, such as the UAU-SIS model that we here consider, the maximal reduction is the trivial partition where all the species are in the same block [4, 26]. We consider non-degenerate reductions using initial partitions with 3 blocks: $\{AI_1, \ldots, AI_N\}$,

[3] Further discussion on the choice of α and β is beyond the scope of this manuscript and we refer the interested reader to [28].

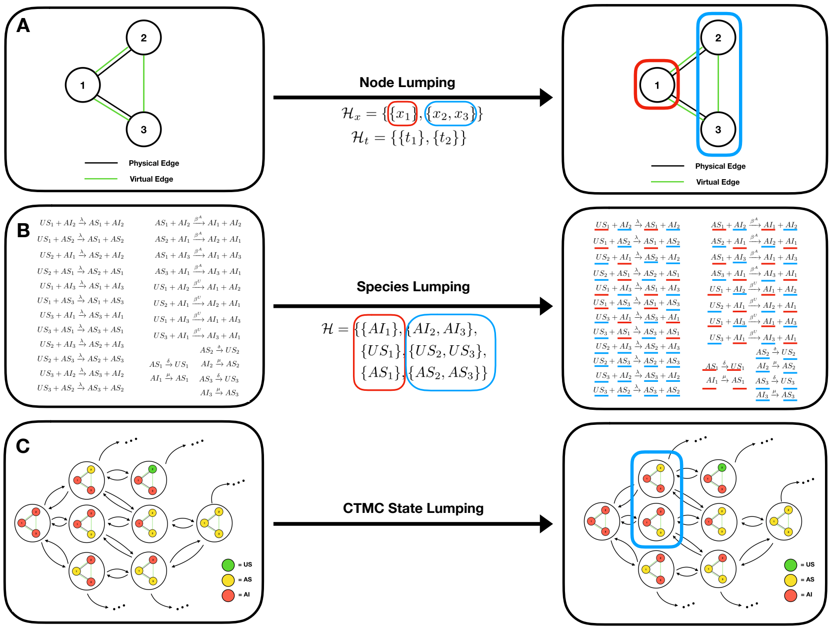
A
1
2
3
Physical Edge
Virtual Edge
Node Lumping
$\mathcal{H}_x = \{\{x_1\}, \{x_2, x_3\}\}$
$\mathcal{H}_t = \{\{t_1\}, \{t_2\}\}$
B
$US_1 + AI_2 \xrightarrow{\lambda} AS_1 + AI_2$
$US_1 + AS_2 \xrightarrow{\lambda} AS_1 + AS_2$
$US_2 + AI_1 \xrightarrow{\lambda} AS_2 + AI_2$
$US_2 + AS_1 \xrightarrow{\lambda} AS_2 + AS_1$
$US_1 + AI_3 \xrightarrow{\lambda} AS_1 + AI_3$
$US_1 + AS_3 \xrightarrow{\lambda} AS_1 + AS_3$
$US_3 + AI_1 \xrightarrow{\lambda} AS_3 + AI_1$
$US_3 + AS_1 \xrightarrow{\lambda} AS_3 + AS_1$
$US_2 + AI_3 \xrightarrow{\lambda} AS_2 + AI_3$
$US_2 + AS_3 \xrightarrow{\lambda} AS_2 + AS_3$
$US_3 + AI_2 \xrightarrow{\lambda} AS_3 + AI_2$
$US_3 + AS_2 \xrightarrow{\lambda} AS_3 + AS_2$
$AS_1 + AI_2 \xrightarrow{\beta^A} AI_1 + AI_2$
$AS_2 + AI_1 \xrightarrow{\beta^A} AI_2 + AI_1$
$AS_1 + AI_3 \xrightarrow{\beta^A} AI_1 + AI_3$
$AS_3 + AI_1 \xrightarrow{\beta^A} AI_3 + AI_1$
$US_1 + AI_2 \xrightarrow{\beta^U} AI_1 + AI_2$
$US_2 + AI_1 \xrightarrow{\beta^U} AI_2 + AI_1$
$US_1 + AI_3 \xrightarrow{\beta^U} AI_1 + AI_3$
$US_3 + AI_1 \xrightarrow{\beta^U} AI_3 + AI_1$
$AS_2 \xrightarrow{\delta} US_2$
$AS_1 \xrightarrow{\delta} US_1$
$AI_2 \xrightarrow{\mu} AS_2$
$AI_1 \xrightarrow{\mu} AS_1$
$AS_3 \xrightarrow{\delta} US_3$
$AI_3 \xrightarrow{\mu} AS_3$
Species Lumping
$\mathcal{H} = \{\{AI_1\}, \{AI_2, AI_3\}, \{US_1\}, \{US_2, US_3\}, \{AS_1\}, \{AS_2, AS_3\}\}$
C
CTMC State Lumping
= US
= AS
= AI

Fig. 1: (A) A compact bi-dimensional representation of a multiplex MLN with physical layer edges (black) and virtual layer edges (green). We show a possible Node Equivalent partition lumping nodes. (B) The UAU-SIS spreading process on the MLN presented in (A). We show the Backward Equivalence lumping species of the Reaction Network. We highlight the equivalent species based on the Node Equivalence presented in (A). (C) A subset of the CTMC state-space (transaction rates omitted) induced by the spreading process from (B). We highlight two states that can be lumped by inheriting the partitions presented in steps (A-B).

$\{AS_1, \ldots, AS_N\}$ and $\{US_1, \ldots, US_N\}$. These experiments showcase the limits in terms of potential reduction size.

Throughout this Section we will refer as *PL-x* the synthetised MLNs with *x* being the number of nodes. In all the experiments that involve runs of the stochastic simulations we used time horizon $T = 10$.

3.1 Experimental setup

All the experiments presented in this paper rely on three components. First, a Python script using the *Networkx* package [13] is used to generate the synthetic Power-Law single-layer networks. Then, both the Power-Law networks and the real-world networks are transformed via a Python script to the desired MLNs by adding the percentage of new edges in the virtual layer. The second step is comprised of a variety of MATLAB scripts. The main purpose of the MATLAB scripts is to parse the instances we generated into a series of models for ERODE [6]. ERODE, a state-of-the-art model reduction tool, is used to compute Backward Equivalence, Forward Equivalence, Stochastic

Equivalence and Node Equivalence. ERODE provides the following outputs: the partitions that were computed and the reduced models (in a format suitable to run SSA simulations). We use the state-of-the-art tool *StochKit* [24] to run the stochastic. All experiments have been conducted on a MacBook Pro with a 2.6 GHz Intel Core i7 with 16 GB of RAM.[4]

3.2 Size of the reduction

In this set of experiments, we compare the size of the obtained reductions using BE, FE and SE, applied to the MLN by translating it to a reaction network formalism, as explained in Sec. 2.4. The results are presented in Table 1 for maximal aggregation. For each instance, we report the percentage of added edges in the virtual layer, the number of species $|S|$ of the Reaction Network of the original model (i.e., $|S| = 3 \cdot N$, where N is the number of nodes), the number of species of the reduced model via BE and the reduction ratio (number of species in the reduced model, divided by the number of species in the original model). Analogously, in the last three columns we present the size of the reductions obtained with FE, SE, as well as their reduction ratio.

First, we notice that FE and SE compute the same partitions: this is because FE and SE both characterise ordinary lumpability and in the case of the UAU-SIS spreading process that we consider in this work they coincide. A general characterisation of a class of spreading processes over MLNs for which the FE and SE partition coincide is beyond the scope of this manuscript and we leave it to future work.

Secondly, we can notice how BE and FE are notions that are not comparable with each other, as discussed in Section 2 and in [8, 4]. However, in some cases, for instance in *EgoFB* with no added edges they might coincide.

We showcase the experimental results on multiplex MLNs in which the physical layer corresponds to the virtual layer without adding any extra edge because this can be interpreted as having two spreading processes over a single layer network. This setting allows us to observe that on the real-world instances where the physical and virtual layer are identical, the partitions obtained with FE and SE correspond to the partitions obtained with SE in [4]. Moreover, we observe that on those instances, the partitions obtained with BE correspond to the partitions obtained labeled as *orbit partitions* in [4]. This is in line with previous results that relate orbit partitions [26], automorphisms [26], bisimulation [21], exact role assignment [27] and equitable partitions in single layer networks. In the presented instances, adding more edges results in more refinement. This is expected to happen because most of the techniques presented exploit symmetries and the act of adding more edges usually leads to a smaller amount of symmetries in the model. Notice how, whilst this intuitive statement is true for most of the instances presented, there are some cases in which adding random edges leads to coarser partitions (for instance, *PL-100* with 10% or 20% added edges). From this set of experiments we observe that the reduction ratio, except in the case of the *EgoFB* [20] instance, does not look appealing. Ideally, the smaller the reduction ratio the better but, when dealing with instances that arise from real-world scenarios, it is known that it is rare to find significant reductions due to their highly non-symmetrical nature. However, we will later

[4] The code and examples are available https://github.com/stefanotognazzi/LumpingForMLNs

show in Section 3.4 that, even with the reduction ratios presented in Table 1, we obtain significant speed-ups in computing stochastic simulations.

Table 1: Size of reductions

Maximal Aggregation		Summary of reductions					
Instance	Added	$\|S\|$	$\|S\|$ (BE)	BE Ratio	$\|S\|$ (FE)	$\|S\|$ (SE)	FE/SE ratio
PL-100	0	300	234	78.0%	258	258	86.0%
PL-100	5	300	255	85.0%	264	264	88.0%
PL-100	10	300	270	90.0%	282	282	94.0%
PL-100	20	300	270	90.0%	276	276	92.0%
PL-500	0	1500	1113	74.2%	1302	1302	86.8%
PL-500	5	1500	1203	80.2%	1302	1302	86.8%
PL-500	10	1500	1296	86.4%	1389	1389	92.6%
PL-500	20	1500	1341	89.4%	1392	1392	92.8%
PL-1000	0	3000	2271	75.7%	2631	2631	87.7%
PL-1000	5	3000	2475	82.5%	2706	2706	90.2%
PL-1000	10	3000	2553	85.1%	2763	2763	92.1%
PL-1000	20	3000	2745	91.5%	2856	2856	95.2%
PL-5000	0	15000	10863	72.4%	12888	12888	85.9%
PL-5000	5	15000	11883	79.2%	13137	13137	87.6%
PL-5000	10	15000	12519	83.5%	13509	13509	90.1%
PL-5000	20	15000	13518	90.1%	14091	14091	93.9%
PL-10000	0	30000	20340	67.8%	24570	24570	81.9%
PL-10000	5	30000	22497	75.0%	25332	25332	84.4%
PL-10000	10	30000	24138	80.5%	26268	26268	87.6%
PL-10000	20	30000	26445	88.2%	27450	27450	91.5%
EgoFB [20]	0	8664	105	1.2%	105	105	1.2%
EgoFB	5	8664	420	4.8%	939	939	10.8%
EgoFB	10	8664	759	8.8%	1659	1659	19.1%
EgoFB	20	8664	1527	17.6%	2949	2949	34.0%
As2000 [19]	0	19422	11070	57.0%	11655	11655	60.0%
As2000	5	19422	13017	67.0%	13455	13455	69.3%
As2000	10	19422	14580	75.1%	14853	14853	76.5%
As2000	20	19422	16584	85.4%	16719	16719	86.1%
PGP [1]	0	32040	23832	74.4%	26019	26019	81.2%
PGP	5	32040	26445	82.5%	27708	27708	86.5%
PGP	10	32040	28221	88.1%	28992	28992	90.5%
PGP	20	32040	30300	94.6%	30597	30597	95.5%

3.3 Cost of the reduction

In this set of experiments, we show the computational cost (in terms of time) of obtaining the reductions. Results are summarised in Table 2. We show for each instance the number of species in the original model ($|S|$) and for each of the proposed techniques the time (in seconds) required by ERODE to obtain the partitions presented in Table 1. BE and FE are computationally efficient. SE is polynomial but, because of the added constraints, in practice it is more computationally costly. In the application we present in this paper, the results show that the preferred option is to use FE to compute the partition that corresponds to the ordinary lumpable of the underlying CTMC.

Table 2: Time of reductions

Instance	Added	$\|S\|$	BE(s)	FE(s)	SE(s)	Instance	Added	$\|S\|$	BE(s)	FE(s)	SE(s)
PL-100	0	300	0.005	0.006	0.025	PL-10000	0	30000	0.705	0.774	324.061
PL-100	5	300	0.004	0.005	0.029	PL-10000	5	30000	0.638	0.687	304.188
PL-100	10	300	0.005	0.005	0.035	PL-10000	10	30000	0.663	0.797	289.318
PL-100	20	300	0.005	0.006	0.031	PL-10000	20	30000	0.688	0.724	275.042
PL-500	0	1500	0.029	0.030	0.419	EgoFB	0	8664	0.094	0.111	0.262
PL-500	5	1500	0.027	0.033	0.400	EgoFB	5	8664	0.098	0.120	1.145
PL-500	10	1500	0.028	0.034	0.470	EgoFB	10	8664	0.097	0.175	2.092
PL-500	20	1500	0.027	0.034	0.474	EgoFB	20	8664	0.123	0.140	4.346
PL-1000	0	3000	0.143	0.146	4.053	As2000	0	19422	0.833	0.901	114.832
PL-1000	5	3000	0.066	0.070	1.624	As2000	5	19422	0.399	0.632	96.712
PL-1000	10	3000	0.096	0.128	2.447	As2000	10	19422	0.471	0.733	175.666
PL-1000	20	3000	0.081	0.105	2.687	As2000	20	19422	0.410	0.702	218.138
PL-5000	0	15000	0.326	0.299	49.818	PGP	0	32040	0.857	1.110	374.756
PL-5000	5	15000	0.289	0.349	59.171	PGP	5	32040	0.884	1.073	409.404
PL-5000	10	15000	0.290	0.399	87.038	PGP	10	32040	0.942	1.043	498.678
PL-5000	20	15000	0.342	0.330	85.778	PGP	20	32040	0.983	1.061	576.374

3.4 Speeding up stochastic simulations

In this set of experiments, we show the benefits of using the reduced models in terms of the speed-up of the stochastic simulations. The results are summarised in Table 3. In this work, we conduct simulations using SSA. All the results are presented in seconds and the reported time of one run is obtained as the time of a run averaged over a repetition of 5 runs. For each instance, we show the time (in seconds) of computing one run of SSA on the full model. In the middle columns, we report the time of computing one run of SSA algorithm for BE, FE and SE reduced models with maximal aggregation. As mentioned in Section 3.2 the reduction ratio presented in Table 1 is only a rough indicator of the potential speed up of the computation. In spite of the fact that the reduction ratios are around 80% of the original model the speed-up is more considerable. For example, let us consider the case of instance *PL-10000* with 5% of added random edges: from Table 1 we can see that with BE we have a reduction ratio of 75% while the computational cost of a single SSA run drops from 514.460 seconds for the original model to 193.087 seconds for the reduced model. In practice, in spite of the reduction ratio being 75%, the cost of a single run on the reduced model has been measured to be 37.5% of the same computation on the original model. This holds for the majority of the instances presented, and can be explained by the run-time of simulation superlinear wrt. node count. These type of speed-ups are inherent to the problem at hand, which is a better speed-up than what has been shown on similar problems, such as centrality-preserving model reduction, where the speed-up of the computation is linear with respect to the size of the reduction (i.e., a reduction ratio of 75% translates to a reduced procedure to calculate the centrality that takes approximately 75% of the time it takes to calculate the same measure on the original model).

Table 3: Time of simulation

		Time for one SSA run (s)						Time for one SSA run (s)			
Instance	Added	Full	BE	FE	SE	Instance	Added	Full	BE	FE	SE
PL-100	0	0.070	0.055	0.061	0.058	PL-10000	0	480.597	141.831	193.790	211.023
PL-100	5	0.066	0.058	0.064	0.064	PL-10000	5	514.460	193.087	230.441	239.246
PL-100	10	0.064	0.059	0.063	0.062	PL-10000	10	535.360	232.286	264.743	267.900
PL-100	20	0.070	0.064	0.067	0.069	PL-10000	20	544.585	291.213	311.796	322.926
PL-500	0	0.321	0.242	0.279	0.286	EgoFB	0	47.974	0.031	0.032	0.031
PL-500	5	0.333	0.269	0.289	0.284	EgoFB	5	48.002	0.102	0.259	0.264
PL-500	10	0.335	0.290	0.311	0.313	EgoFB	10	48.194	0.192	0.742	0.793
PL-500	20	0.347	0.313	0.329	0.332	EgoFB	20	48.591	0.579	2.613	2.855
PL-1000	0	0.932	0.696	0.764	0.813	As2000	0	211.622	34.123	42.825	43.195
PL-1000	5	0.916	0.745	0.826	0.830	As2000	5	215.274	55.684	64.782	66.634
PL-1000	10	0.934	0.787	0.842	0.867	As2000	10	214.124	80.212	87.279	87.460
PL-1000	20	0.969	0.874	0.912	0.904	As2000	20	218.152	129.608	133.483	138.029
PL-5000	0	12.503	7.942	9.696	10.106	PGP	0	66.687	42.880	51.821	54.000
PL-5000	5	12.873	9.395	10.304	10.317	PGP	5	68.481	51.766	56.884	59.885
PL-5000	10	13.057	10.042	10.938	12.311	PGP	10	70.543	58.104	63.297	65.535
PL-5000	20	13.667	11.723	12.525	12.872	PGP	20	73.802	68.135	69.810	74.214

3.5 Approximation of the reduction

In this set of experiments we aim at providing experimental evidence that, despite the fact that BE in general is an approximation of the stochastic semantics, we can use that reduction in this scenario as a good approximation of the original solution. We use synthetised MLNs with 15% of added edges in the virtual layer with a number of nodes ranging from 20 to 200. In order to obtain some precision in the solution we computed 1 Million runs of SSA on each of the presented instances. Accuracy results are presented with respect to the simulated number of nodes in each state and compared to the solution of the original instance. In this set of experiments we fix an initial partition based on the initial conditions proposed in [11] (i.e., we start with 20% of the nodes in the AI state). In Table 4 we present the average across all runs of the number of nodes in each state at the end of the simulation (i.e., at time $t = 10$). In Table 5 we show the maximum error, in terms of percentage of distance to the solution of the original instance, that we could observe at all time points. From [4] we know that the reduction obtained via FE/SE is exact in the sense that it can be used to replicate exactly the stochastic semantics of the original model. Therefore, any discrepancy in the results between the full original model and the FE/SE reductions presented in Table 4 is either due to numerical approximations or to the stochastic nature of the simulations. Although a preliminary analysis showed that 1 Million runs was a reasonable number of simulations in order to obtain a small standard deviation in the results of the sample, it can still account for the main reason of the discrepancy between the measures presented.

Table 4: Approximations of reduction

Realistic	Accuracy			Realistic	Accuracy		
PL-20	AI	US	AS	PL-70	AI	US	AS
Full	1.0516	17.0271	1.9213	Full	14.6530	33.7650	21.5820
BE	1.0532	17.0223	1.9245	BE	14.6513	33.7610	21.5877
FE	1.0545	17.0196	1.9260	FE	14.6539	33.7509	21.5952
SE	1.0529	17.0237	1.9234	SE	14.6474	33.7628	21.5898
PL-30	AI	US	AS	PL-80	AI	US	AS
Full	3.2101	21.5280	5.2620	Full	12.7763	47.6297	19.5940
BE	3.2082	21.5209	5.2709	BE	12.7791	47.6390	19.5819
FE	3.2113	21.5217	5.2670	FE	12.7897	47.6148	19.5955
SE	3.2079	21.5219	5.2702	SE	12.7873	47.6241	19.5887
PL-40	AI	US	AS	PL-90	AI	US	AS
Full	2.6129	32.8668	4.5204	Full	11.6084	59.9792	18.4124
BE	2.6187	32.8548	4.5265	BE	11.5948	59.9885	18.4167
FE	2.6154	32.8568	4.5278	FE	11.6079	59.9708	18.4213
SE	2.6188	32.8537	4.5275	SE	11.6004	59.9825	18.4172
PL-50	AI	US	AS	PL-100	AI	US	AS
Full	7.5197	30.8025	11.6778	Full	19.5981	52.0239	28.3779
BE	7.5192	30.8018	11.6790	BE	19.6470	51.9174	28.4356
FE	7.5202	30.8059	11.6739	FE	19.5975	52.0339	28.3685
SE	7.5289	30.7930	11.6781	SE	19.5944	52.0302	28.3754
PL-60	AI	US	AS	PL-200	AI	US	AS
Full	7.4002	41.0514	11.5483	Full	27.2781	130.9980	41.7239
BE	7.3935	41.0569	11.5496	BE	27.2851	130.9977	41.7172
FE	7.3923	41.0700	11.5376	FE	27.2913	130.9824	41.7262
SE	7.3948	41.0664	11.5389	SE	27.2903	130.9894	41.7203

Table 5: Max error

Instance	BE
PL-20	0.36%
PL-30	0.17%
PL-40	0.22%
PL-50	0.04%
PL-60	0.16%
PL-70	0.05%
PL-80	0.06%
PL-90	0.15%
PL-100	0.25%
PL-200	0.03%

3.6 Reduction at the level of nodes

In this set of experiments we show how to obtain the same partition obtainable with BE lumping at the level of the reaction network by using Node Equivalence (NE) from Section 2.7 that acts at the level of the MLN's nodes. We show experimental evidence that the two reductions coincide when interpreted over the nodes. A general characterisation of a class of spreading processes over MLNs for which the BE and NE partition coincide is beyond the scope of this manuscript and we leave it to future work. The results are summarised in Table 6. We report the results for the Power-Law MLNs with 5% of added edges. To facilitate the interpretation of the results we provide in the table, alongside the number of original species and reduced species, the size in terms of nodes of the MLNs of the obtained partitions.

Table 6: Node Equivalence, 5% added edges

Maximal aggregation	Node Lumping				Species Lumping			
Instance	$\|S\|$	$\|S\|$(NE)	Nodes (NE)	Time (s)	$\|S\|$ full	$\|S\|$ (BE)	Nodes (BE)	Time (s)
PL-100	102	87	85	0.003	300	255	85	0.004
PL-500	502	403	401	0.011	1500	1203	401	0.031
PL-1000	1002	827	825	0.028	3000	2475	825	0.096
PL-5000	5002	3963	3961	0.124	15000	11883	3961	0.300
PL-10000	10002	7501	7499	0.369	30000	22497	7499	0.656
PL-50000	50002	38027	38025	1.818	150000	114075	38025	3.896
PL-100000	100002	77489	77487	3.921	300000	232461	77487	7.388
PL-500000	500002	382068	382066	24.049	1500000	1146198	382066	59.850
PL-1000000	1000002	776457	776455	56.244	3000000	——— O.O.M. ———-		

For example, the first row of the table showcases the instance *PL-100* which has 100 nodes. The number of species in the original model is 102 (i.e., 100 nodes plus one species per layer). The number of reduced species is 87 and the species that encode the layers do not get lumped together, therefore the total number of blocks when considering the nodes is $87 - 2 = 85$. This calculation is straightforward for the reduction at the Reaction Network level, for each node we have 3 Species and therefore in order to translate it at the node level we need to divide by 3. For example, for *PL-100* we have 300 as the number of species in the original model, then we obtain 255 species in the reduced model and we obtain to how many node they correspond with $255/3 = 85$. The advantage of this approach is that Node Equivalence reduces a model that has a number of species that is $|V| + |L|$, while, using BE on the reaction network we need to reduce a system such that $|S| = 3 \cdot |V|$. From a theoretical perspective this fact does not yield any improvement in terms of complexity but we show in Table 6 that in practice we obtain a speed up and the ability to scale to larger size networks.

4 Conclusions and Future Works

Stochastic semantics are a key tool to understand and study spreading processes in networked systems. Analysing interacting spreading processes on complex multiplex Multi-Layer Networks is computationally costly, if feasible at all. In this work, we extended a variety of lumping-based automated model reduction techniques to interacting spreading processes on Multiplex Multi-Layer Networks that allows the modeler to run the stochastic simulations at a cheaper computational cost. Our findings show experimental evidence that in the context of multispread processes over MLNs, efficient reduction techniques originally designed to exactly preserve differential semantics, faithfully abstract the stochastic semantics. In future work, we plan to investigate these results from a theoretical standpoint, that will set foundations for exploiting this scalable approach in practice.

References

1. Marián Boguñá, Romualdo Pastor-Satorras, Albert Diaz-Guilera, and Alex Arenas. Models of social networks based on social distance attachment. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 70:056122, 12 2004.
2. Piotr Bródka, Katarzyna Musial, and Jaroslaw Jankowski. Interacting spreading processes in multilayer networks: A systematic review. *IEEE Access*, 8:10316–10341, 2020.
3. Peter Buchholz. Exact and ordinary lumpability in finite markov chains. *J. Applied Probability*, 31:59–74, 1994.
4. Luca Cardelli, Isabel Cristina Perez-Verona, Mirco Tribastone, Max Tschaikowski, Andrea Vandin, and Tabea Waizmann. Exact maximal reduction of stochastic reaction networks by species lumping. *Bioinformatics*, 02 2021. btab081.
5. Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Symbolic computation of differential equivalences. In *43st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, 2016.
6. Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. ERODE: A tool for the evaluation and reduction of ordinary differential equations. In *TACAS*, pages 310–328, 2017.
7. Manlio De Domenico, Albert Solè-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason Porter, Sergio Gomez, and Alex Arenas. Mathematical formulation of multi-layer networks. *Physical Review X*, 3, 07 2013.
8. Jérôme Feret, Thomas Henzinger, Heinz Koeppl, and Tatjana Petrov. Lumpability abstractions of rule-based systems. *Theoretical Computer Science*, 431:137–164, 2012.
9. Kelly Finn, Matthew Silk, Mason Porter, and Noa Pinter-Wollman. The use of multilayer network analysis in animal behaviour. *Animal Behaviour*, 149:7–22, 03 2019.
10. Arnab Ganguly, Tatjana Petrov, and Heinz Koeppl. Markov chain aggregation and its applications to combinatorial reaction networks. *Journal of mathematical biology*, 69(3):767–797, 2014.
11. Clara Granell, Sergio Gomez, and Alex Arenas. Dynamical interplay between awareness and epidemic spreading in multiplex networks. *Physical review letters*, 111:128701, 09 2013.
12. Gerrit Grossmann and Luca Bortolussi. *Reducing Spreading Processes on Networks to Markov Population Models*, pages 292–309. 09 2019.
13. Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.
14. Z. L. Hu, L. Wang, and C. B. Tang. Locating the source node of diffusion process in cyber-physical networks via minimum observers. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(6):063117, 2019.
15. John G Kemeny and J Laurie Snell. Finite continuous time markov chains. *Theory of Probability & Its Applications*, 6(1):101–105, 1961.
16. Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 07 2014.
17. Jérôme Kunegis. KONECT – The Koblenz Network Collection. In *Proc. Int. Conf. on World Wide Web Companion*, pages 1343–1350, 2013.
18. Thomas G Kurtz. Solutions of ordinary differential equations as limits of pure jump markov processes. *Journal of applied Probability*, 7(1):49–58, 1970.
19. Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1):2–es, March 2007.

20. Jure Leskovec and Julian Mcauley. Learning to discover social circles in ego networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
21. M. Marx and M. Masuch. Regular equivalence and dynamic logic. *Social Networks*, 25(1):51 – 65, 2003.
22. Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. *Rev. Mod. Phys.*, 87:925–979, Aug 2015.
23. Tatjana Petrov and Stefano Tognazzi. Centrality-preserving exact reductions of multi-layer networks. In *Leveraging Applications of Formal Methods, Verification and Validation: Engineering Principles*, pages 397–415, Cham, 2020. Springer International Publishing.
24. Kevin R. Sanft, Sheng Wu, Min Roh, Jin Fu, Rone Kwei Lim, and Linda R. Petzold. StochKit2: software for discrete stochastic simulation of biochemical systems with events. *Bioinformatics*, 27(17):2457–2458, 07 2011.
25. Joaquín Sanz, Cheng-Yi Xia, Sandro Meloni, and Yamir Moreno. Dynamics of interacting diseases. *Physical Review X*, 4, 02 2014.
26. Péter Simon, Michael Taylor, and Istvan Kiss. Exact epidemic models on graphs using graph-automorphism driven lumping. *Journal of mathematical biology*, 62:479–508, 04 2011.
27. Stefano Tognazzi, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. Differential equivalence yields network centrality. In *Leveraging Applications of Formal Methods, Verification and Validation. Distributed Systems - 8th International Symposium, ISoLA 2018, Limassol, Cyprus, November 5-9, 2018, Proceedings, Part III*, volume 11246 of *Lecture Notes in Computer Science*, pages 186–201. Springer, 2018.
28. Francesco Tudisco, Francesca Arrigo, and Antoine Gautier. Node and layer eigenvector centralities for multiplex networks. *SIAM Journal of Applied Mathematics*, 78(2):853–876, 2018.
29. Piet Van Mieghem, Jasmina Omic, and Robert Kooij. Virus spread in networks. *IEEE/ACM Trans. Netw.*, 17(1):1–14, February 2009.
30. Xuetao Wei, Nicholas Valler, B. Aditya Prakash, Iulian Neamtiu, Michalis Faloutsos, and Christos Faloutsos. Competing memes propagation on networks: A case study of composite networks. *ACM SIGCOMM Computer Communication Review*, 42:5–11, 10 2012.