Lumping Reductions for Multispread in Multi-Layer Networks

Tatjana Petrov[1,2] and Stefano Tognazzi[1,2]

[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 arise from simple local interaction among a large number of actors through different networks of interactions. Computational modelling and analysis of such phenomena is challenging due to the combinatorial explosion of possible network configurations. Traditional (single layer) networks are commonly used to encode the heterogeneous relationships among agents but are limited to a single type of interaction. Multiplex Multi-Layer networks (MLNs) have been introduced to allow the modeler to compactly and naturally describe multiple types of interactions and multiple simultaneous spreading phenomena. The downside is an increase in the complexity of the already challenging task of the analysis and simulation of such spreading processes. In this paper we explore the use of lumping techniques that preserve dynamics, previously applied to Continuous Time Markov Chains (CTMC) and single layer networks to multiple spreading processes on MLNs.

Keywords: Multiplex 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 actors, 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 [14], to understanding animal collectives [7] and online social networks [20] to cite a few.

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) [19, 9, 2]. For instance, a network of n actors where each actor can either be infected (I) or susceptible (S) to infection, gives rise to 2^n possible network configurations. On the other hand, *Reaction network* formalism facilitates the description of actors 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 actors in state S or I: (i) $S + I \rightarrow 2I$ for infection spread

and (ii) $I \to S$ for recovery. Such model typically does not specify whether two actors 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 actors 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 actor 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 actors in a network, e.g. $S_i + I_j \to I_i + I_j$, and conditioning the respective rate to the existence of an edge between actors [9, 2, 8]. Moreover, in reality, the same group of individuals can partake in different spreading processes at different interaction networks, which in turn affect each-other [1]. 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 [8]. 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 actors, as well as inter-layer correlations, in a natural and compact way [8, 15, 5, 12] . However, more detailed description further challenges the respective computational simulation and analysis. Formal reductions based on lumping states that are behaviourally equivalent are desirable [17], 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 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.

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 $V_p \subseteq \{\mathtt{x}_1, \mathtt{x}_2, \ldots\}$, 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 $\mathtt{x}_{H,1}, \mathtt{x}_{H,2}, \ldots, \mathtt{x}_{H,|H|}$. 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 actors 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 the *stochastic* semantics. It is worth mentioning when we refer to the deterministic semantics of reaction networks we use Ordinary Differential Equations with mass-action kinetics.

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,

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

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

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

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.

2.2 Multiplex Multi-Layer Networks

In this paper, we use 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 actors [5]. This paper will focus on *undirected* multiplex networks.

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

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

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*:*

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

that is, $\mathcal{A}_{ijl}$ represents the presence of an edge between actors i and j on layer l.

2.3 Lumping species in a reaction network

We next review three formal reductions techniques for lumping species, based on the reaction network description [3]. 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[3] [3].
- (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 $\sim_{FE}$.
- (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 [2]. Finding the partition $\sim_{SE}$ is polynomial in the size of reaction network $O(|R||S|log|S|)$ This significantly improves the complexity of searching for the lumpable partition directly over the expanded CTMC.

2.4 Lumping actors in multiplex networks

Lumping techniques can be used as an efficient algorithmic procedure to compute structural properties of actors within a network. In recent works [15], we use formal reductions to compute one notion of eigenvector centrality for multiplex MLNs, proposed

[3] The algorithm is a variation of the Paige-Tarjan algorithm

in [18]. The centrality is defined through a 2-map, $\boldsymbol{f}$*-eigenvector centrality*, in which the first component of the map represents the centrality associated to the *actors*, while the second component is centrality associated to the *layers*.

Definition 2. *([18])* 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 an actor i is a sum of the centralities of each of its neighbouring actors, weighted by the product of the edge-weight and the centrality of the layer at which that connection lies. The parameters α and β are introduced in order to guarantee convergence and respectively well-definedness in case of undirected MLNs. [4] In [18] the centrality vector of the actors 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.

- (Actor Equivalence) $\sim_{AE} \subseteq V_N \times V_N$ is an actor equivalence, where two actors are lumped, if they have the same $\boldsymbol{f}$-eigenvector centrality value (i.e., $x_i \sim_{AE} x_j$, if $x_i^* = x_j^*$). Finding the partition $\sim_{AE}$ 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 [15].

Related Works In [11] the authors propose a unified taxonomy of MLN simplification techniques. It is worth mentioning that FE, BE and SE can not be considered as explicit MLN simplification techniques, because they are acting directly at the level of the *Reaction Network*. However, AE is a MLN simplification, because it is acting at the level of actors (it is an *aggregation* technique based on *positional equivalence*).

3 Results

3.1 Interacting spreading processes on Multiplex Multi-Layer Networks

In this work we focus on interacting spreading processes, sometimes referred as *multispread* processes.

Definition 3. *Let $\mathcal{G}$ be a multiplex. A multispread $\mathcal{MS} = (\mathcal{G}, IS, P, IC, SR, LIR)$ is a tuple composed by:*

- *A multiplex $\mathcal{G}$ with N actors and L layers;*
- *A set of internal states $IS = \{IS1, \ldots, ISM\}$;*
- *A set of rates $P = \{r_1, \ldots, r_R\}$ s.t. for all $i \in \{1, \ldots, R\}$, $r_i \in \mathbb{R}^+$;*
- *A set of initial conditions IC, $IC : \{1, \ldots, N\} \to IS$;*

[4] Further discussion on the choice of α and β is beyond the scope of this manuscript and we refer the interested reader to [18].

- *A set of single actor rules SR. SR rules are in the form $ISx \xrightarrow{r} ISy$, where $ISx, ISy \in IS$ and $r \in P$.*
- *A set of local interaction rules LIR. LIR rules are in the form $ISx \overset{l}{—} ISy \xrightarrow{r} ISy \overset{l}{—} ISy$ where $ISx, ISy \in IS$, $l \in \{1, \ldots, L\}$ and $r \in P$.*

Where $ISx \overset{l}{—} ISy$ represents the existence of an edge in layer l between agents that are in internal state ISx and ISy respectively.

Definition 4. *(Compiling a multispread into a RN)*
Let $\mathcal{MS} = (\mathcal{G}, IS, P, IC, SR, LIR)$ be a multispread process, the resulting Reaction Network $RN_{\mathcal{MS}} = (S, R)$ is constructed as follows:

- *The set of species S is the union, for all $i \in \{1, \ldots, N\}$ of the set of all the internal states of each actor $Act_i = \{IS1_i, \ldots, ISM_i\}$.*
- *The set of reactions $R = R_{SR} \cup R_{LIR}$ is the union of the reactions that we compile from the single actor rules R_{SR} and the local interaction rules R_{LIR}.*
- *For each rule $sr \in SR$ where sr is in the form $ISx \xrightarrow{r} ISy$ we build the set of reactions $R_{sr} = \{ISx_i \xrightarrow{r} ISy_i \mid i \in \{1, \ldots, N\}\}$*
- *For each rule $lir \in LIR$ where lir is in the form $ISx \overset{l}{—} ISy \xrightarrow{r} ISy \overset{l}{—} ISy$ we build the set of reactions $R_{lir} = \{ISx_i + ISy_j \xrightarrow{r} ISy_i + ISy_j \mid (i, j) \in E^{(l)}\}$.*
- *We set the following initial conditions:*

$$ISx_i = \begin{cases} 1 \text{ if } IC(i) = ISx, \\ 0 \text{ if } IC(i) \neq ISx. \end{cases}$$

3.2 The multispread model

In this work, we evaluate the performance of the lumping techniques reviewed in Section 2 to multispreads that arise from MLNs. We will use two different sets of benchmarks: real-world networks from the Koblenz Network Collection [13], and a set of synthetic networks. The synthetic networks will have a physical layer (referred as layer 1) built using a power-law degree distribution network generated with a configuration model with exponent 2.5. In all case studies, the virtual layer (referred as layer 2) 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. In this work we use the following multispread inspired by the interacting spreading processes presented in [8].

$$\begin{aligned} IS =& \{US, AS, AI\} \\ P =& \{\delta = 0.6, \mu = 0.4, \beta^A = 0.01, \beta^U = 0.4, \lambda = 0.15\} \\ SR =& \{AS \xrightarrow{\delta} US, AI \xrightarrow{\mu} AS\} \\ LIR =& \{AS \overset{1}{—} AI \xrightarrow{\beta^A} AI \overset{1}{—} AI, US \overset{1}{—} AI \xrightarrow{\beta^U} AI \overset{1}{—} AI, \\ & US \overset{2}{—} AI \xrightarrow{\lambda} AS \overset{2}{—} AI, US \overset{2}{—} AS \xrightarrow{\lambda} AS \overset{2}{—} AS\} \end{aligned}$$

We consider non-degenerate reductions using initial partitions with 3 blocks: $\{AI_1, \ldots, AI_N\}$, $\{AS_1, \ldots, AS_N\}$ and $\{US_1, \ldots, US_N\}$.

3.3 Experimental setup

All the experiments presented in this paper rely on three components. First, a Python script using the *Networkx* package [10] is used to generate the synthetic Power-Law MLNs. The second step is comprised of MATLAB scripts which parse instances into a series of models for ERODE [4]. ERODE, a state-of-the-art model reduction tool, is used to compute Backward Equivalence, Forward Equivalence, Stochastic Equivalence and Actor Equivalence. ERODE provides the following outputs: the partitions that were computed and the reduced models. We use the state-of-the-art tool *StochKit* [16] 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.[5] Throughout this Section we will use *PL-x* to refer to the synthetised MLNs with *x* being the number of actors. In all the experiments that involve runs of the stochastic simulations we used time horizon $T = 10$.

3.4 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, using Def. 4. The results are presented in Table 1. 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 actors), 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.

Secondly, we can notice how BE and FE are notions that are not comparable with each other, as discussed in [6, 2].

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. 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 show in Section 3.6 that, even with the reduction ratios presented in Table 1, we obtain significant speed-ups in computing stochastic simulations.

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

Table 1: Size of reductions

Maximal Aggregation			Summary of reductions				Maximal Aggregation			Summary of reductions			
Instance	Added	$\|S\|$	$\|S\|$ (BE)	BE Ratio	$\|S\|$ (FE/SE)	FE/SE ratio	Instance	Added	$\|S\|$	$\|S\|$ (BE)	BE Ratio	$\|S\|$ (FE/SE)	FE/SE ratio
PL-100	5	300	255	85.0%	264	88.0%	PL-10000	5	30000	22497	75.0%	25332	84.4%
PL-100	10	300	270	90.0%	282	94.0%	PL-10000	10	30000	24138	80.5%	26268	87.6%
PL-500	5	1500	1203	80.2%	1302	86.8%	EgoFB	5	8664	420	4.8%	939	10.8%
PL-500	10	1500	1296	86.4%	1389	92.6%	EgoFB	10	8664	759	8.8%	1659	19.1%
PL-1000	5	3000	2475	82.5%	2706	90.2%	As2000	5	19422	13017	67.0%	13455	69.3%
PL-1000	10	3000	2553	85.1%	2763	92.1%	As2000	10	19422	14580	75.1%	14853	76.5%
PL-5000	5	15000	11883	79.2%	13137	87.6%	PGP	5	32040	26445	82.5%	27708	86.5%
PL-5000	10	15000	12519	83.5%	13509	90.1%	PGP	10	32040	28221	88.1%	28992	90.5%

3.5 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.

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	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-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-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-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

3.6 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. We conduct stochastic 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. This can be explained by the fact that the run-time of simulation is superlinear wrt. actor count.

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	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-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-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-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

3.7 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 actors ranging from 20 to 200. In order to obtain 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 actors 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 [8]. In Table 4 we present the average across all runs of the number of actors in each state at the end of the simulation. 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 [2] 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, we report the results obtained using BE.

3.8 Reduction at the level of actors

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 Actor Equivalence (AE) from Section 2.4 that acts at the level of the MLN's actors. We show experimental evidence that the two reductions coincide when interpreted over the actors. 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 actors of the MLNs of the obtained partitions.

The advantage of this approach is that Actor 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.

Table 4: Approximations of reduction

Instance	Accuracy			Instance	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
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
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
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
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

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%

Table 6: Actor Equivalence, 5% added edges

Maximal aggregation	Actor Lumping				Species Lumping			
Instance	$\|S\|$	$\|S\|$(AE)	Actors (AE)	Time (s)	$\|S\|$ full	$\|S\|$ (BE)	Actors (BE)	Time (s)
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. ———		

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.

Acknowledgements This work was supported by the Ministry of Science, Research and the Arts of the state of Baden-Württemberg, and the DFG Centre of Excellence 2117 'Centre for the Advanced Study of Collective Behaviour' (ID: 422037984). The authors would like to thank Giacomo Rapisardi for the inspiring discussions on the topic and Andrea Vandin for the support and the insights on the use of the tool ERODE.

References

1. Piotr Bródka, Katarzyna Musial, and Jaroslaw Jankowski. Interacting spreading processes in multilayer networks: A systematic review. *IEEE Access*, 8:10316–10341, 2020.
2. 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.
3. 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.
4. 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.
5. 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.
6. Jérôme Feret, Thomas Henzinger, Heinz Koeppl, and Tatjana Petrov. Lumpability abstractions of rule-based systems. *Theoretical Computer Science*, 431:137–164, 2012.
7. 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.
8. Clara Granell, Sergio Gomez, and Alex Arenas. Dynamical interplay between awareness and epidemic spreading in multiplex networks. *Physical review letters*, 111:128701, 09 2013.
9. Gerrit Grossmann and Luca Bortolussi. *Reducing Spreading Processes on Networks to Markov Population Models*, pages 292–309. 09 2019.
10. 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.
11. Roberto Interdonato, Matteo Magnani, Diego Perna, Andrea Tagarelli, and Davide Vega. Multilayer network simplification: Approaches, models and methods. *Comput. Sci. Rev.*, 36:100246, 2020.
12. 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.
13. Jérôme Kunegis. KONECT – The Koblenz Network Collection. In *Proc. Int. Conf. on World Wide Web Companion*, pages 1343–1350, 2013.
14. Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. *Rev. Mod. Phys.*, 87:925–979, Aug 2015.
15. 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.
16. 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.

17. 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.
18. 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.
19. Piet Van Mieghem, Jasmina Omic, and Robert Kooij. Virus spread in networks. *IEEE/ACM Trans. Netw.*, 17(1):1–14, February 2009.
20. 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.