J. Appl. Prob. **56,** 647–671 (2019)
*doi:*10.1017/jpr.2019.44
© *Applied Probability Trust* 2019

APPROXIMATE LUMPABILITY FOR MARKOVIAN AGENT-BASED MODELS USING LOCAL SYMMETRIES

WASIUR R. KHUDABUKHSH,* *The Ohio State University*
ARNAB AUDDY,** *Columbia University*
YANN DISSER,*** AND
HEINZ KOEPPL,**** *Technische Universität Darmstadt*

Abstract

We study a Markovian agent-based model (MABM) in this paper. Each agent is endowed with a local state that changes over time as the agent interacts with its neighbours. The neighbourhood structure is given by a graph. Recently, Simon, Taylor, and Kiss [40] used the automorphisms of the underlying graph to generate a lumpable partition of the joint state space, ensuring Markovianness of the lumped process for binary dynamics. However, many large random graphs tend to become asymmetric, rendering the automorphism-based lumping approach ineffective as a tool of model reduction. In order to mitigate this problem, we propose a lumping method based on a notion of local symmetry, which compares only local neighbourhoods of vertices. Since local symmetry only ensures approximate lumpability, we quantify the approximation error by means of the Kullback–Leibler divergence rate between the original Markov chain and a *lifted* Markov chain. We prove the approximation error decreases monotonically. The connections to fibrations of graphs are also discussed.

Keywords: Lumpability; aggregation; agent-based models; local symmetry; fibration

2010 Mathematics Subject Classification: Primary 60J28
Secondary 60J22; 60J27; 65C50

1. Introduction

In this paper, a Markovian agent-based model (MABM) refers to a stochastic interacting particle system (IPS) with a *finite* local state space. Given a graph with N vertices, we endow each vertex with a local state that varies over time stochastically as the vertex interacts with its neighbours. Let $G=(V, E)$ be a graph (possibly a realization of a random graph), where $V:=\{1, 2, \dots, N\}$ is the set of vertices and $E\subseteq V\times V$ is the set of edges. For simplicity, we assume G is undirected in the sense that $(u, v)\in E$ whenever $(v, u)\in E$, for $u, v\in V$, and all characteristics of $(u, v)\in E$ are same as those of (v, u). In other words, having an

Received 5 April 2018; revision received 20 December 2018.

* Postal address: Mathematical Biosciences Institute, The Ohio State University, Jennings Hall, 3rd Floor, 1735 Neil Avenue, Columbus, Ohio 43210, USA. Email address: khudabukhsh.2@osu.edu

** Postal address: Department of Statistics, Columbia University, Room 1005 SSW, MC 4690, 1255 Amsterdam Avenue, New York, NY 10027, USA. Email address: arnab.auddy@columbia.edu

*** Postal address: Department of Mathematics, Technische Universität Darmstadt, Dolivostrasse 15, 64293 Darmstadt, Germany. Email address: disser@mathematik.tu-darmstadt.de

**** Postal address: Department of Electrical Engineering and Information Technology, Technische Universität Darmstadt, Rundeturmstrasse 12, 64283 Darmstadt, Germany. Email address: heinz.koeppl@bcs.tu-darmstadt.de

undirected edge between two vertices u and v is assumed to be equivalent to having two identical directed edges between u to v in opposite directions. Later in this paper, this approach will be useful for studying connections between our methods and some notion of morphisms between directed graphs. Let $X_i(t)$ denote the local state of vertex $i \in V$ at time $t \in \mathcal{T} := [0, T]$ for some $T > 0$. For simplicity, we assume the vertices have the same finite local state space $\mathcal{X} := \{1, 2, \dots, K\}$ for some positive integer K. We assume the process $X := (X_1, X_2, \dots, X_N) \in \mathcal{X}^N$ is a continuous-time Markov chain (CTMC), whose transition intensities depend on G. The objective of this paper is to devise an approximately lumpable partition of $\mathcal{X}^N$ [10, 15, 25, 37, 38] using local symmetries of the graph G.

Recently, Simon, Taylor, and Kiss [40] introduced a novel lumping procedure based on the automorphisms of the underlying graph G. They considered a stochastic susceptible-infected-susceptible (SIS) epidemic process on a graph. They showed that when the automorphism group is known, a lumpable partition can be obtained by determining the orbits of the elements of the state space with respect to the automorphism group. The idea of lumping using graph automorphisms is innovative. However, it is not always efficient for two reasons. First, finding all automorphisms without additional information about the graph structure is computationally prohibitive, especially for large graphs (see [4]). Second, there may be too few automorphisms to engender significant state space reduction [40] as many large random graphs tend to be asymmetric with high probability (see [27, 31, 32]). Therefore, we propose a lumping procedure based on a *local* notion of symmetry [16] taking into account only local (k-hop) neighbourhoods of each vertex. In our approach, we construct an equitable partition [21, Chapter 9] of V by clubbing together vertices that are locally symmetric. We say two vertices u and v are locally symmetric if there exists an isomorphism f between their respective local neighbourhoods (the induced subgraphs) such that $f(u) = v$. This is less restrictive than the existence of an automorphism g on the entire graph G mapping u to v.

Local symmetry-driven lumping allows for a more profitable aggregation than automorphism. Therefore, even when there are too few automorphisms, we can still achieve significant state space reduction by means of local symmetry-driven lumping. The price we pay for this gain is that the resultant lumped process will only be *approximately* Markovian. We quantify the approximation error in terms of the Kullback–Leibler (KL) divergence rate between the uniformization of the original process X and a Markov chain *lifted* from the lumped process (details are provided in Section 5). For a particular type of lifting called π-lifting, we prove that the approximation error decreases as we increase the number of hops in our consideration of local symmetries.

Interestingly, the equivalence classes of the local symmetry can be shown to be the same as the fibres of a graph fibration [9]. Therefore, the fibres can also be used to aggregate the states of $\mathcal{X}^N$ to achieve approximate lumpability in the same fashion as we do with local symmetry. In addition to that, the problem of finding a lumpable partition for our MABM shares interesting connections with other related concepts in algebraic graph theory, such as colour refinement for directed graphs [3, 7] and coverings [1]. A discussion of these connections paves the way for potential application of graph-theoretic algorithms to problems in applied probability, and vice versa.

The paper is structured as follows. In Section 2, we discuss some mathematical preliminaries required for the rest of the paper. We formally introduce the MABM in Section 3. The lumping based on graph automorphisms and related results are presented in Section 4. In Section 5, we extend the lumping ideas to local symmetry of graphs. Connections to graph fibrations are explored in Section 6. The approximation error associated with local symmetry-driven lumping is studied in Section 7. Our theoretical discussions are also complemented with some numerical

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

results on Erdős–Rényi, Barabási–Albert preferential attachment, and Watts–Strogatz small-world graphs. Finally, we conclude the paper with a short discussion in Section 8.

2. Mathematical preliminaries

Notational conventions. We use $\mathbb{N}$ and $\mathbb{R}$ to denote the set of natural numbers and the set of real numbers. Also, we define $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ and $\mathbb{R}_+ := \mathbb{R} \setminus (-\infty, 0]$. Additionally, we denote the set $\{1, 2, \ldots, N\}$ by $[N]$. For a set A, we denote its cardinality by $|A|$, and the class of all subsets of A by 2^A. Given $N, K \in \mathbb{N}$, the set of all non-negative integer solutions to the Diophantine equation $x_1 + x_2 + \cdots + x_K = N$ is denoted by $\Lambda(N, K)$, that is,

$$\Lambda(N, K) := \{x = (x_1, x_2, \ldots, x_K) \in \mathbb{N}_0{}^K \mid x_1 + x_2 + \cdots + x_K = N\}.$$

We use $\mathbb{1}(.)$ to denote the indicator function. The symmetric group on a set A is denoted by $\mathsf{Sym}(A)$.

2.1. Lumpability

We first define (strong) lumpability for a discrete time Markov chain (DTMC) for ease of understanding. Standard references on this topic are [10, 25, 37, 38].

Let $\{Y(t)\}_{t\in\mathbb{N}}$ be a DTMC on a state space $\mathcal{Y} = [K]$ with transition probability matrix $T = ((t_{i,j}))_{K\times K}$, where $t_{i,j} := \mathsf{P}(Y(2) = j \mid Y(1) = i)$. Given a partition $\{\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_M\}$ of $\mathcal{Y}$, we define a process $\{Z(t)\}_{t\in\mathbb{N}}$ on $[M]$ as follows: $Z(t) = i \in [M]$ if and only if $Y(t) \in \mathcal{Y}_i$, for each $t \in \mathbb{N}$. The process Z is called the *lumped* or the *aggregated* process. The sets $\mathcal{Y}_i$ are often called lumping classes.

Definition 1. (*Lumpability of a DTMC.*) A DTMC Y on a state space $\mathcal{Y}$ is lumpable with respect to the partition $\{\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_M\}$ of $\mathcal{Y}$, if the lumped process Z is itself a DTMC for every choice of the initial distribution of Y [25, Chapter VI, p. 124].

A necessary and sufficient condition for lumpability, known as Dynkin's criterion in the literature, is the following: for any two pairs of lumping classes $\mathcal{Y}_i$ and $\mathcal{Y}_j$ with $i \neq j$, the transition probabilities of moving into $\mathcal{Y}_j$ from any two states in $\mathcal{Y}_i$ are the same, i.e. $t_{u,\mathcal{Y}_j} = t_{v,\mathcal{Y}_j}$ for all $u, v \in \mathcal{Y}_i$, where we have used the shorthand notation $t_{u,A} = \sum_{j\in A} t_{u,j}$ for $A \subseteq \mathcal{Y}$. The common values, i.e. $\tilde{t}_{i,j} = t_{u,\mathcal{Y}_j}$, for some $u \in \mathcal{Y}_i$, and $i, j \in [M]$, form the transition probabilities of the lumped process Z. Let $\tilde{T} = ((\tilde{t}_{i,j}))_{M\times M}$. Since Dynkin's criterion is both necessary and sufficient, some authors alternatively define lumpability in terms of Dynkin's criterion. In the literature, the process Z is sometimes denoted as $Z = \mathsf{agg}(Y)$.

Now, we move to the continuous-time case. The lumped process and the notion of lumpability for the continuous-time case are defined similarly.

Definition 2. (*Lumpability of a CTMC.*) A CTMC Y on a state space $\mathcal{Y}$ is lumpable with respect to the partition $\{\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_M\}$ of $\mathcal{Y}$, if the lumped process Z is itself a CTMC for every choice of the initial distribution of Y.

The lumpability of a CTMC can be equivalently described in terms of lumpability of a linear system of ordinary differential equations (ODEs). Consider the linear system $\dot{y} = yA$, where $A = ((a_{i,j}))$ is a $K \times K$ matrix (representing the transition rate matrix of the corresponding continuous-time Markov chain).

Definition 3. (*Lumpability of a linear system.*) The linear system $\dot{y} = yA$ is said to be lumpable with respect to a partition $\{\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_M\}$ of $\mathcal{Y}$, if there exists an $M \times M$ matrix $B = ((b_{i,j}))$ satisfying Dynkin's criterion, i.e. if $b_{i,j} = \sum_{l\in\mathcal{Y}_j} a_{u,l} = \sum_{l\in\mathcal{Y}_j} a_{v,l}$ for all $u, v \in \mathcal{Y}_i$.

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

An alternative approach to study the lumpability of a CTMC is via its uniformization. This approach will be particularly useful when we discuss lumpability using local symmetries later. Let us consider a CTMC $\{Y(t)\}_{t\in\mathcal{T}}$ with transition rate matrix $A=((a_{i,j}))$. It is known that the original CTMC is lumpable with respect to a given partition if and only if the uniformized DTMC is lumpable with respect to the same partition. The uniformized DTMC $\tilde{Y}$ is often denoted by $\mathsf{unif}(Y)$, i.e. $\tilde{Y}=\mathsf{unif}(Y)$. It was proved in [19, 38] that

$$\mathsf{agg}(\mathsf{unif}(Y))=\mathsf{unif}(\mathsf{agg}(Y)).$$

Another useful observation that will be helpful later is regarding permutation of the states. It is intuitive that permutation of elements of the state space does not destroy the lumpability property of a process. The proof of the following remark is straightforward, but is provided in Appendix A for the sake of completeness.

Remark 1. Let Y be a CTMC on $\mathcal{Y}$ with transition rate matrix $A=((a_{i,j}))$. Let $f\in\mathsf{Sym}(\mathcal{Y})$ be used to permute the states. If Y (or the linear system $\dot{y}=yA$) is lumpable with respect to a partition $\{\mathcal{Y}_1,\mathcal{Y}_2,\ldots,\mathcal{Y}_M\}$, then the process $Z=f(Y)$ is lumpable with respect to the partition $\{\tilde{\mathcal{Y}}_1,\tilde{\mathcal{Y}}_2,\ldots,\tilde{\mathcal{Y}}_M\}$, where $\tilde{\mathcal{Y}}_i=\{f(u)\mid u\in\mathcal{Y}_i\}$.

The notion of lumpability considered here is often referred to as strong lumpability in the literature. There is also a notion of weak lumpability [37, 38], which we do not consider. Therefore, the term ‘lumpability’ always refers to strong lumpability in this paper. Also, note that both the DTMC and the CTMC, and the lumped processes in Definitions 1, and 3 are all assumed time-homogeneous. However, in general, it is possible to extend the notion of lumpability to allow the lumped process to be Markovian, but not necessarily time-homogeneous.

3. Markovian agent-based model

3.1. Interaction rules and the transition intensities

The most important ingredients of an MABM are the interaction rules of the agent-based local processes X_i. These rules of interaction determine the dynamics of the process. Note that an MABM can also be viewed as a collection of local CTMCs that are connected to each other via the graph G. In other words, each X_i can be seen as a local CTMC, conditioned on the rest. In this work, we assume the intensities of the local CTMC X_i depend on the local states X_j of the neighbours of the vertex $i\in V$ (such that $(i,j)\in E$). Let $d_i=|\{j\in V\mid (i,j)\in E\}|$ denote the number of neighbours of vertex i. Additionally, we assume the intensities depend only on the counts of neighbours for each local state $a\in\mathcal{X}$. Therefore, we define the following summary function c that returns population counts for different configurations of local states,

$$c\colon \{\varnothing\}\cup\left(\bigcup_{l=1}^{N}\mathcal{X}^l\right)\longrightarrow\{\varnothing\}\cup\left(\bigcup_{l=1}^{N}\Lambda(l,K)\right)$$

such that, for $x=(x_1,x_2,\ldots,x_l)\in\mathcal{X}^l$, and $l\in[N]$,

$$c(x)=(y_1,y_2,\ldots,y_K)\in\Lambda(l,K)\quad\text{where } y_i=|\{x_j=i\in\mathcal{X}: j=1,2,\ldots,l\}|,$$

and set $c(\varnothing)=\varnothing$. Here, x is a given configuration of states of l vertices and y_i is the count of vertices (within the given configuration x) that are in the ith state. The empty set $\varnothing$ is

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

used to denote the neighbourhood of an isolated vertex. An important feature of the set-valued function c is that it is permutation-invariant in the sense that $c(x)=c(x')$ if the elements of x' are permutations of the elements of x. In order to extract the neighbourhood information out of the global configuration, we define a family of set-valued functions n_i in the following way:

$$n_i\colon \mathcal{X}^N \longrightarrow \{\varnothing\} \cup \left(\bigcup_{l=1}^{N-1} \mathcal{X}^l\right) \quad \text{for } i \in [N],$$

such that, for $x=(x_1, x_2, \ldots, x_N) \in \mathcal{X}^N$,

$$n_i(x)=\begin{cases} (x_{i_1}, x_{i_2}, \ldots, x_{i_l}) & \text{if } (i, i_j) \in E \text{ for all } j=1, 2, \ldots, l \text{ and } l=d_i, \\ \varnothing & \text{otherwise.} \end{cases}$$

The neighbourhood extraction function n_i for the ith vertex takes as input x, the (global) configuration of the states of all N vertices, and returns the tuple of states of the neighbours of the vertex i. Having defined these two important functions, we now define the interaction rules by means of local transition intensities. We assume the intensities depend only on the type of local transition and the summary of the neighbourhood configuration of a vertex. Therefore, we define the local intensity function

$$\gamma\colon \ \mathcal{X} \times \mathcal{X} \times \left(\{\varnothing\} \cup \left(\bigcup_{l=1}^{N-1} \Lambda(l, K)\right)\right) \longrightarrow \mathbb{R}_+, \tag{3.1}$$

where we interpret $\gamma(a, b, y)$ as the local intensity of making a transition from local state a to b by a vertex when the summary of its neighbourhood configuration is y.

We are now in a position to specify the transition rate or the infinitesimal generator matrix for our MABM X. Note that the process X jumps from a state x to y whenever one of the local processes X_i jumps. Therefore, only one of the coordinates of the states x and y differ. Let the $K^N \times K^N$ matrix $Q=((q_{x,y}))$ denote the transition rate matrix of X. The elements of the matrix Q are given by

$$q_{x,y}=\begin{cases} \sum_{i\in[N]} \mathbb{1}\,(x_i \neq y_i, x_j=y_j\ \forall j \in V \setminus \{i\})\, \gamma(x_i, y_i, c(n_i(x))) & \text{if } x \neq y, \\ -\sum_{y\neq x} q_{x,y} & \text{if } x=y. \end{cases}$$

We interpret $q_{x,y}$ as the rate of transition from x to y, where $x, y \in \mathcal{X}^N$. For ease of understanding, we have suffixed the entries of Q by the different configurations $x, y \in \mathcal{X}^N$ and interpret them as functions on $\mathcal{X}^N \times \mathcal{X}^N$, instead of introducing a bijection between $\mathcal{X}^N$ and $[K^N]$ to label the states in a linear order so that the suffixes range over the integers from 1 to K^N. Note that the particular choice of bijection to label the states is immaterial for our purposes, because such a bijection essentially yields a permutation of $[K^N]$, and in the light of Proposition 1, does not alter lumpability properties of Q. Finally, we study the dynamics of X via the linear system

$$\dot{p}=pQ.$$

The vector-valued function p gives the probability distribution of X.

3.2. Examples

Susceptible-infected-susceptible (SIS) epidemics. The SIS epidemic model [40] captures the dynamics of an epidemic spread over a human or an animal population. It encapsulates

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

binary dynamics in the sense that the local state space is written as $\mathcal{X} := \{1, 2\}$, where 1 indicates susceptibility and 2 indicates an infected status. Infected vertices infect one of its randomly chosen neighbours at each ticking of a Poisson clock with a fixed rate $a > 0$. Infected vertices themselves recover to susceptibility at a rate $b \geq 0$, independent of the neighbours' statuses. When $b = 0$, the model is called a susceptible-infected (SI) model. The local transition intensities are given by

$$\gamma(1, 2, (x_1, x_2)) = x_2 a \quad \text{and} \quad \gamma(2, 1, (x_1, x_2)) = b.$$

We set γ to zero in every other case. This fully describes the dynamics of the system.

Peer-to-peer live media streaming systems. Peer-to-peer networks are engineered networks where the vertices, called peers, communicate with each other to perform certain tasks in a distributed fashion. In particular, content delivery platforms such as BitTorrent, file-sharing platforms such as Gnutella, and (live) media (audio/video) streaming platforms use peer-to-peer networks. The main advantage of peer-to-peer systems over centralized systems is their 'scalability'. That is, the performance of a peer-to-peer system scales well with the number of peers by virtue of the distributed task-sharing aspect, even though the global demand increases as more and more peers join the network. For the purposes of performance analysis, Markov chain models are often used for such systems.

In a peer-to-peer live-streaming system, each peer maintains a buffer of length L. The availability of a media chunk at buffer index $i \in [L]$ is indicated by 1, and likewise unavailability is indicated by 0 (see [26]). Therefore, the local state space is given by $\mathcal{X} = \{0, 1\}^L$. Set $K = 2^L$ so that $\{0, 1\}^L$ can be put in one-to-one correspondence with $[K]$. The chunk at buffer index L, if available, is played back at rate unity and then removed. After playback, all other chunks are moved one index to the right, that is, the chunk at buffer index i is shifted to buffer index $i + 1$. The central server selects a finite number of peers at random and uploads chunks at buffer index 1. All other peers (not receiving chunks from the server) download chunks from their neighbours, following a *pull* mechanism. Note that there are also systems where the peers *push* chunks into their neighbours' buffers instead of pulling. However, we do not consider them here. The peers contact their neighbours to download missing chunks. The neighbours fulfil the request if the requested chunk is available. When multiple chunks are missing, the peers prioritize the chunks in some way, giving rise to different chunk selection strategies, such as the latest deadline first (LDF) and the earliest deadline first (EDF) strategies. Let us introduce a function, called the chunk selection function, that captures this prioritization, usually represented as probabilities. Let $s\colon [L] \times \mathcal{X} \times \mathcal{X}$ be the chunk selection function. We interpret $s(i, u, v)$ as the probability of a vertex with buffer configuration u selecting to fill buffer index i when it contacts a neighbour with buffer configuration v.

We assume the peer-to-peer live-streaming system described above is Markovian. That is, the peers are assumed to maintain their private Poisson clocks at the tickings of which they contact their neighbours for missing chunks. Let the rate of these Poisson clocks be $a > 0$. Let $y_1, y_2, \ldots, y_K$ be a linear arrangement of the states in $\mathcal{X}$. Denote the jth component of y_i by $y_{i,j}$, i.e. $y_i = (y_{i,1}, y_{i,2}, \ldots, y_{i,L})$. The local intensity function is then given by [26],

$$\gamma(u, u + e_j, (x_1, x_2, \ldots, x_K)) = a \sum_{i \in [K]} \mathbb{1}(y_{i,j} = 1)\, x_i s(j, u, y_i) \quad \text{if } j > 1,$$

where e_j is the jth unit vector in the L-dimensional Euclidean space, and $(x_1, x_2, \ldots, x_K)$ is the population count vector of the neighbours of a vertex with different buffer configurations.

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

Apart from the above transitions due to download of a chunk from a neighbour, there are two other transitions, namely, the transition due to the shifting after playback that takes place at rate unity irrespective of the buffer configurations of the neighbours, and the transition due to being directly served by the server. The latter event also takes place irrespective of the buffer configurations of the neighbours, but a rate that depends on the exact implementation set-up of the peer-to-peer system. See [26] for a detailed account.

4. Automorphism-based lumping of an MABM

Now we discuss how graph automorphisms can be used to lump states of X. The idea was introduced by Simon, Taylor, and Kiss [40] for SIS epidemics on graphs. The purpose of lumping states is to generate a Markov chain on a smaller state space. However, we should make sure that the loss of information is not too much. For instance, X is always lumpable with respect to the partition $\{\mathcal{X}^N\}$, but if all states are lumped together, all information about the dynamics of X is lost except for the fact that total probability is conserved at all times. On the other hand, X is also lumpable with respect to the partition $\{\{x\} \mid x \in \mathcal{X}^N\}$, which retains all the information but does not yield any state space reduction. Therefore, one needs to find a meaningful partition that yields as much state space reduction as possible with minimal loss of information. For an MABM, population counts are very useful quantities. Therefore, in order to retain information about the population counts, we first partition $\mathcal{X}^N$ into $\{\mathcal{X}_a \mid a \in \Lambda(N, K)\}$, that is,

$$\mathcal{X}^N = \cup_{a\in\Lambda(N,K)}\mathcal{X}_a \quad \text{where } \mathcal{X}_a := \{b \in \mathcal{X}^N \mid c(b) = a\}, \tag{4.1}$$

and then seek a lumpable partition that is ideally *minimally* finer than this. The partition in (4.1) lumps together states that produce the same population counts. The size of this partition, i.e. $|\{\mathcal{X}_a \mid a \in \Lambda(N, K)\}|$, is $\binom{N+K-1}{K-1}$. Note that, in the standard mean-field approach, one assumes that X is lumpable with respect to the partition in (4.1) and studies (approximate) master equations (Kolmogorov forward equations) corresponding to the different population counts. Next, we refine this partition using automorphisms.

A bijection $f\colon V \longrightarrow V$ is an *automorphism* on G if $(i, j) \in E$ if and only if $(f(i), f(j)) \in E$, for all $i, j \in V$ (see [21]). The collection of all automorphisms forms a group under the composition of maps. This group is denoted by $\mathsf{Aut}(G)$. Clearly, $\mathsf{Aut}(G)$ is a subgroup of $\mathsf{Sym}(V)$. In order to use automorphisms to produce a partition of $\mathcal{X}^N$, we shall let $\mathsf{Aut}(G)$ act on $\mathcal{X}^N$. We define the following group action (a map from $\mathsf{Aut}(G) \times \mathcal{X}^N$ to $\mathcal{X}^N$):

$$f \cdot x = y \in \mathcal{X}^N \quad \text{if and only if } x_{f(i)} = y_i \text{ for all } i \in [N], \quad \text{for } f \in \mathsf{Aut}(G),\ x \in \mathcal{X}^N.$$

The rationale is that, for our purpose, an automorphism needs to preserve the local states of vertices as well. Note that the action of the group $\mathsf{Aut}(G)$ defined above can be used to introduce an equivalence relation on $\mathcal{X}^N$ as follows: we say x and y are equivalent with respect to the action of $\mathsf{Aut}(G)$, denoted as $x \sim y$, if and only if there exists an $f \in \mathsf{Aut}(G)$ such that $f \cdot x = y$. The equivalence classes $\{\tilde{\mathcal{X}}_1, \tilde{\mathcal{X}}_2, \ldots, \tilde{\mathcal{X}}_M\}$ of the relation $\sim$ yield a lumpable partition of $\mathcal{X}^N$. Moreover, the partition thus obtained is finer than $\{\mathcal{X}_a \mid a \in \Lambda(N, K)\}$. We prove this in the following.

Proposition 1. *The partition $\{\tilde{\mathcal{X}}_1, \tilde{\mathcal{X}}_2, \ldots, \tilde{\mathcal{X}}_M\}$ induced by the equivalence relation $\sim$, i.e. the quotient space $\mathcal{X}^N/\sim$, is a refinement of $\{\mathcal{X}_a \mid a \in \Lambda(N, K)\}$. That is, for each $i \in [M]$, there exists an $a \in \Lambda(N, K)$ such that $\tilde{\mathcal{X}}_i \subseteq \mathcal{X}_a$.*

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

Proof. Pick any $\tilde{\mathcal{X}}_i$ and $x \in \tilde{\mathcal{X}}_i$. Then $a = c(x) \in \Lambda(N, K)$, and therefore $x \in \mathcal{X}_a$. The proof is completed when we show that every other y in $\tilde{\mathcal{X}}_i$ is also in $\mathcal{X}_a$. Now $y \in \tilde{\mathcal{X}}_i$ implies $x \sim y$, and therefore there exists an $f \in \mathsf{Aut}(G)$ such that $f \cdot x = y$. From the permutation invariance of c, we get $c(y) = c(f \cdot x) = c(x) = a$, implying $y \in \mathcal{X}_a$. □

Theorem 1. *The CTMC X with transition rate matrix Q (or equivalently the linear system* $\dot{p} = pQ$*) is lumpable with respect to the quotient space* $\mathcal{X}^N / \sim$*, the partition* $\{\tilde{\mathcal{X}}_1, \tilde{\mathcal{X}}_2, \ldots, \tilde{\mathcal{X}}_M\}$ *induced by the equivalence relation* $\sim$*.*

Before proving Theorem 1, we prove the following useful lemma regarding the neighbourhood function and the action of the group $\mathsf{Aut}(G)$.

Lemma 1. *For all* $i \in [N]$ *and for any* $z \in \mathcal{X}^N$*, the following is true for all* $f \in \mathsf{Aut}(G)$*:*

$$n_{f^{-1}(i)}(f \cdot z) = n_i(z). \tag{4.2}$$

Proof of Lemma 1. Let us put $f \cdot z = x$ and $f^{-1}(i) = k$. If $d_k = 0$, the assertion follows immediately because both sides of (4.2) are the empty set. Therefore, we assume $d_k = l > 0$. Then,

$$\begin{aligned} n_{f^{-1}(i)}(f \cdot z) &= (x_{i_1}, x_{i_2}, \ldots, x_{i_l}) \quad \text{if } (i_j, k) \in E \text{ for all } j \in [i_l] \\ &= (z_{f(i_1)}, z_{f(i_2)}, \ldots, z_{f(i_l)}) \quad \text{if } (i_j, k) \in E \text{ for all } j \in [i_l] \\ &= n_{f(k)}(z), \end{aligned}$$

but $f(k) = i$, implying $n_{f^{-1}(i)}(f \cdot z) = n_i(z)$. □

Now we present the proof of Theorem 1.

Proof of Theorem 1. We check Dynkin's criterion to establish lumpability. For any two distinct $i, j \in [M]$, we check if $\tilde{q}_{i,j} = \sum_{y \in \tilde{\mathcal{X}}_j} q_{x,y} = \sum_{y \in \tilde{\mathcal{X}}_j} q_{z,y}$ for each distinct pair $x, z \in \tilde{\mathcal{X}}_i$. Since $z \sim x$, there exists an $f \in \mathsf{Aut}(G)$ such that $f \cdot z = x$. The idea is to apply f on the states of $\tilde{\mathcal{X}}_j$ and then show that, for any two states $x, z \in \tilde{\mathcal{X}}_i$, there are two states $y, f \cdot y \in \tilde{\mathcal{X}}_j$ such that the neighbourhood information is preserved, that is,

$$\begin{aligned} \sum_{y \in \tilde{\mathcal{X}}_j} q_{x,y} &= \sum_{y \in \tilde{\mathcal{X}}_j} \sum_{i \in [N]} \mathbb{1}\,(x_i \neq y_i, x_j = y_j \ \forall j \neq i)\, \gamma(x_i, y_i, c(n_i(x))) \\ &= \sum_{f \cdot y \in \tilde{\mathcal{X}}_j} \sum_{i \in [N]} \mathbb{1}\,(x_i \neq y_{f(i)}, x_j = y_{f(j)} \ \forall j \neq i)\, \gamma(x_i, y_{f(i)}, c(n_i(x))) \\ &= \sum_{f \cdot y \in \tilde{\mathcal{X}}_j} \sum_{i \in [N]} \mathbb{1}\,(z_{f(i)} \neq y_{f(i)}, z_{f(j)} = y_{f(j)} \ \forall j \neq i)\, \gamma(z_{f(i)}, y_{f(i)}, c(n_i(f \cdot z))) \\ &= \sum_{f \cdot y \in \tilde{\mathcal{X}}_j} \sum_{f^{-1}(i) \in [N]} \mathbb{1}\,(z_i \neq y_i, z_j = y_j \ \forall j \neq i)\, \gamma(z_i, y_i, c(n_i(z))) \\ &= \sum_{y \in \tilde{\mathcal{X}}_j} \sum_{i \in [N]} \mathbb{1}\,(z_i \neq y_i, z_j = y_j \ \forall j \neq i)\, \gamma(z_i, y_i, c(n_i(z))) \\ &= \sum_{y \in \tilde{\mathcal{X}}_j} q_{z,y}, \end{aligned}$$

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

where we have used $n_{f^{-1}(i)}(f \cdot z) = n_i(z)$ from Lemma 1. Denoting the common value by $\tilde{q}_{i,j} = \sum_{y \in \tilde{\mathcal{X}}_j} q_{x,y}$, the matrix $\tilde{Q} = ((\tilde{q}_{i,j}))$ is the transition rate matrix of $\mathsf{agg}(X)$. □

Remark 2. From the perspective of group theory, finding the lumping classes is equivalent to determining the orbits of states in $\mathcal{X}^N$ with respect to the group $\mathsf{Aut}(G)$. For a state $x \in \mathcal{X}^N$, the orbit of x with respect to the action of the group $\mathsf{Aut}(G)$, denoted as $\mathsf{Aut}(G) \cdot x$, is defined by $\mathsf{Aut}(G) \cdot x = \{f \cdot x \mid f \in \mathsf{Aut}(G)\}$.

Example 1. (*Complete graph.*) The automorphism group $\mathsf{Aut}(G)$ for the complete graph is $\mathsf{Sym}([N])$. Therefore, any two states $x, y \in \mathcal{X}^N$ can be lumped together if y is a rearrangement of components of x, i.e. $y = f \cdot x$ for some $f \in \mathsf{Sym}([N])$. As a consequence, $\{\mathcal{X}_a \mid a \in \Lambda(N, K)\}$ itself is a lumpable partition of $\mathcal{X}^N$.

Example 2. (*Star graph.*) An automorphism on a star graph leaves the central node (root) unchanged and permutes the rest of the nodes (leaf nodes) in any possible manner. Without loss of generality, let us assume the central node is labelled N. Then, the automorphism group $\mathsf{Aut}(G)$ is given by

$$\begin{aligned}&\mathsf{Aut}(G) = \\ &\quad \{g \in \mathsf{Sym}([N]) \mid g(N) = N,\, g(i) = f(i) \text{ for all } i \in [N-1], \text{ for some } f \in \mathsf{Sym}([N-1])\}.\end{aligned}$$

Example 3. (*Cycle graph.*) The automorphisms of a cycle graphs are the reflections and rotations of the graph, forming a group that is also known as the dihedral group. Therefore, there are $2N$ automorphisms. Simon, Taylor, and Kiss [40] showed that the dihedral group leads to a non-trivial lumping of states.

Example 4. (*Trees.*) For a star graph, we noted that an automorphism permutes the leaves but needs to leave the root unchanged. Similarly, for a tree, we start with the leaves. Any two leaves connected to the same parent node can be freely permuted. However, whenever we permute two leaf nodes that have different parents, we also need to permute the parents to preserve the neighbourhood structure. Therefore, an automorphism on a tree necessarily maps vertices to vertices at the same height.

5. Lumping states using local symmetry

In this section, we discuss lumping ideas based on a local notion of automorphism. In many cases, the number of automorphisms decrease drastically as the graph grows arbitrarily large. For instance, it is known that Erdős–Rényi random graphs tend to be asymmetric with probability approaching unity as the size of the graph N grows to infinity [31]. Similar statements are true for d-regular random graphs under various sets of conditions on d relative to the number of vertices N [27], and random graphs with specified degree distributions [32]. As a consequence, the automorphism-based lumping tends to be ineffective in state space reduction as the size of the graphs grows arbitrarily. Therefore, it is desirable to bring in a notion of local automorphism or local symmetry that would allow swapping vertices that are locally indistinguishable (i.e. have similar neighbourhoods), but are not so globally. This notion of symmetry is weaker than an automorphism, which endows global symmetry on a graph. However, the potential gain is in the ability to engender state space reduction when the graph grows arbitrarily large, rendering automorphism-based lumping virtually ineffective. In the following, we make these ideas precise.

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

5.1. Local symmetry

There have been several attempts to formulate a more flexible notion of local symmetry. However, the literature seems divided on this and there is not a single universally accepted concept. In our set-up, it seems intuitive that two vertices that are locally indistinguishable in a large graph would also behave indistinguishably and could therefore be swapped. A notion of local symmetry identifying such vertices was proposed in [16], which we adopt in this paper. We need a few definitions to make precise what we mean by two vertices being locally indistinguishable.

In order to define locality, we need some notion of distance between vertices of G. Let $d(u, v)$ denote the smallest distance (length of the minimal path) between two vertices $u, v \in V$. If u, and v are not connected, i.e. there is no path between them, we simply set $d(u, v) = \infty$.

Definition 4. Given a vertex $u \in V$, define its k-neighbourhood in G, denoted by $N_k(u)$, as follows:

$$N_k \colon V \longrightarrow 2^V \quad \text{such that } N_k(u) := \{v \in V \mid d(u, v) \leq k\}.$$

Let $G[N_k(u)]$ denote the subgraph of G induced by $N_k(u)$. The notion of locality we adopt in this paper hinges on these k-neighbourhoods and their induced subgraphs. If two vertices induce isomorphic subgraphs, they are indistinguishable locally and we say they are k-locally symmetric [16].

Definition 5. Two vertices $u, v \in V$ are defined to be k-locally symmetric if there exists an isomorphism f between $G[N_k(u)]$ and $G[N_k(v)]$ such that $f(u) = v$.

Therefore, two vertices $u, v \in V$ are k-locally symmetric if their kth-order local structures (k-hop neighbourhoods) are equivalent in the sense that there is a structure-preserving (edge-preserving in this case) bijection between them. When $k = 1$, we simply say the vertices are *locally* symmetric.

As with automorphisms, local symmetries also induce an equivalence relation on the set of vertices V. We say two vertices $u, v \in V$ are equivalent with respect to k-local symmetry, denoted by $u \overset{k}{\sim} v$, if there exists an isomorphism f between $G[N_k(u)]$ and $G[N_k(v)]$ such that $f(u) = v$. The notion of local symmetry is related to the concept of views in the discrete mathematics literature [23, 43]. The view of depth k of a vertex is a tree containing all walks of length k leaving that vertex. However, note that, in our context, the induced subgraphs $G[N_k(u)]$ need not be trees. The following facts about local symmetry are useful for our study of lumpability [16, 35].

Proposition 2. *The following properties are satisfied by k-local symmetry.*

(i) *For $u, v \in V$, $u \overset{k+1}{\sim} v \Longrightarrow u \overset{k}{\sim} v$. Consequently, $V/\overset{k+1}{\sim}$, the equivalence classes of $\overset{k+1}{\sim}$, form a refinement of $V/\overset{k}{\sim}$, the equivalence classes of $\overset{k}{\sim}$.*

(ii) *If the equivalence classes of $\overset{k+1}{\sim}$ are the same as those of $\overset{k}{\sim}$, the equivalence classes of* all $\overset{k+j}{\sim}$ *are the same as those of $\overset{k}{\sim}$, for $j \in \mathbb{N}$.*

(iii) *If $k \geq \operatorname{diam}(G)$, the diameter of G, then, for two vertices $u, v \in V$, we have $u \overset{k}{\sim} v$ if and only if there exists an $f \in \mathsf{Aut}(G)$ such that $f(u) = v$. That is, k-local symmetry is equivalent to automorphism if k is at least as large as the diameter of G.*

In addition to the above, it can be verified that the quotient spaces $V/\overset{k}{\sim}$ are equitable partitions [21, Chapter 9] for each $k \geq 1$. We use these properties to lump states of $\mathcal{X}^N$ in the next section.

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

5.2. Lumping states using local symmetry

The procedure to lump states in $\mathcal{X}^N$ using local symmetry is similar to the procedure used to lump states using automorphism. However, unlike the case with automorphism, we now allow permutations that only need to ensure symmetry locally. That is, in order to lump states using k-local symmetry, we allow permuting two vertices u and v in V if and only if u and v are k-locally symmetric. Therefore, define

$$\Psi_k(G) := \{f \in \mathsf{Sym}(V) \mid f(u) = v \text{ if and only if } u \overset{k}{\sim} v, \text{ for } u, v \in V\}.$$

We refer to $|\Psi_k(G)|$ as the number of k-local symmetries. It can be verified that $\Psi_k(G)$, for each $k \geq 1$, forms a group under the composition of maps. Therefore, we can let the group $\Psi_k(G)$ act on $\mathcal{X}^N$. We define the action of $\Psi_k(G)$ as follows:

$$f \cdot x = y \in \mathcal{X}^N \quad \text{if and only if } x_{f(i)} = y_i \text{ for all } i \in [N], \text{ for } f \in \Psi_k(G), x \in \mathcal{X}^N.$$

Note that a state x in $\mathcal{X}^N$ is taken to y if and only if the local states of all vertices are preserved and two vertices are swapped only when they are k-locally symmetric. The above action induces the following partition of the state space: two states $x, y \in \mathcal{X}^N$ are said to be equivalent with respect to k-local symmetry, denoted as $x \overset{k}{\sim} y$, if there exists an $f \in \Psi_k(G)$ such that $f \cdot x = y$. We use the same symbol $\overset{k}{\sim}$ since there is no scope for confusion. The equivalence classes of $\overset{k}{\sim}$ are obtained, as before, by determining the orbits of states in $\mathcal{X}^N$. The orbit of a state $x \in \mathcal{X}^N$ is given by $\Psi_k(G) \cdot x := \{f \cdot x \in \mathcal{X}^N \mid f \in \Psi_k(G)\}$.

The partition thus obtained (based on k-local symmetry) does not, in general, guarantee lumpability, i.e. the X need to be lumpable with respect to $\mathcal{X}^N/\overset{k}{\sim}$. Therefore, we construct another Markov chain that approximates the lumped process and seek to quantify the approximation error in the next section. The following observation is integral to the quantification of the approximation error incurred when states of $\mathcal{X}^N$ are lumped according to k-local symmetry instead of automorphism.

Proposition 3. *The quotient space $\mathcal{X}^N/\overset{k+1}{\sim}$ is a refinement of $\mathcal{X}^N/\overset{k}{\sim}$.*

Proof of Proposition 3. Let $\mathcal{X}_1^{(k+1)}, \mathcal{X}_2^{(k+1)}, \ldots, \mathcal{X}_{M_{k+1}}^{(k+1)}$ be the equivalence classes of $\overset{k+1}{\sim}$. Also, denote the equivalence classes of $\overset{k}{\sim}$ by $\mathcal{X}_1^{(k)}, \mathcal{X}_2^{(k)}, \ldots, \mathcal{X}_{M_k}^{(k)}$. Let $i \in [M_{k+1}]$ and $x \in \mathcal{X}_i^{(k+1)}$. If $\mathcal{X}_i^{(k+1)}$ is a singleton, the identity map is the only map in $\Psi_{k+1}(G)$, but it is also in $\Psi_k(G)$. Therefore, $x \in \mathcal{X}_j^{(k)}$ for some $j \in [M_k]$, and the assertion follows. If $\mathcal{X}_i^{(k+1)}$ has at least two elements, say, x, y, then $y \overset{k+1}{\sim} x$. By Proposition 2, we must have $y \overset{k}{\sim} x$. Therefore, there exists a $j \in [M_k]$ such that $x, y \in \mathcal{X}_j^{(k)}$. Since the choice of x, y is arbitrary, the assertion follows. □

For practical applications, one would start with $\mathcal{X}^N/\overset{1}{\sim}$ and then iteratively obtain further refinements $\mathcal{X}^N/\overset{2}{\sim}$, $\mathcal{X}^N/\overset{3}{\sim}$, and so on until satisfactory accuracy is achieved (assuming we can quantify accuracy for the time being). In the light of Proposition 2, two important remarks are in order. They emphasize the benefits of local symmetry-driven lumping over automorphism-driven lumping.

Remark 3. In an algorithmic implementation, item (ii) in Proposition 2 provides a stopping rule for an iterative procedure to obtain local symmetry-driven partitions. That is, we can stop at the first instance of no improvement (the equivalence classes of $\overset{k+1}{\sim}$ and $\overset{k}{\sim}$ are the same). For the sake of illustration, consider the chain graph in Figure 1. For this trivial example, notice that going from $k = 1$ to $k = 2$ does not cause any improvement. Therefore, we can stop at $k = 1$ even though the diameter is 4.

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

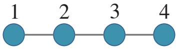

FIGURE 1: A chain graph with $1\overset{1}{\sim}4$ and $2\overset{1}{\sim}3$ as well as $1\overset{2}{\sim}4$ and $2\overset{2}{\sim}3$.

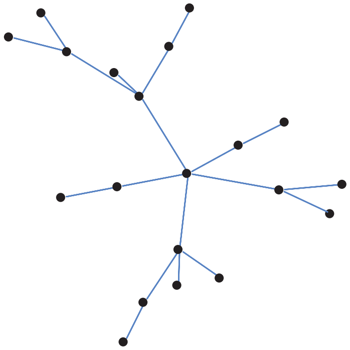

FIGURE 2: A graph with 20 vertices for Example 5.

TABLE 1: State space reduction with k.

k	# of local symmetries	Size of the reduced state space	Compression level
1	2228	14362	98.63%
2	222	68800	93.44%
3	39	215040	79.49%
4	32	248832	76.27%
5	16	276480	73.63%

Remark 4. The diameters in many random graphs grow slowly as the number of vertices goes to infinity. For instance, the diameter of Erdős–Rényi random graphs with N vertices and edge probability λ/N, for some fixed $\lambda > 1$, grows as $\log N$ (see [36]). A similar result holds for Newman, Strogatz, and Watts graphs too (see [12, Lemma 1.2]). In the light of item (iii) of Proposition 2, our approach needs (at most) as many steps as the diameter of G to produce an *exactly* lumpable partition of $\mathcal{X}^N$. Note that $k \geq \operatorname{diam}(G)$ is only a sufficient condition for $\mathcal{X}^N/\overset{k}{\sim}$ to be an exactly lumpable partition. For practical purposes, we actually achieve sufficient accuracy (including exact lumpability) even for small values of $k < \operatorname{diam}(G)$. Supporting numerical results are provided later in Section 7.

In the next example, we show the amount of state space reduction that can be achieved by means of local symmetry-driven lumping.

Example 5. Consider an SIS process on the graph with 20 vertices in Figure 2. The number of states in this case is $2^{20} = 1048576$. However, there are plenty of local symmetries that we can exploit to reduce the size of the state space. In Table 1, we summarize the state space reduction achieved for increasing k. The compression level (used to quantify the reduction achieved) is calculated as one minus the ratio of the size of the reduced state space to the size of the original state space, i.e. 1048576.

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

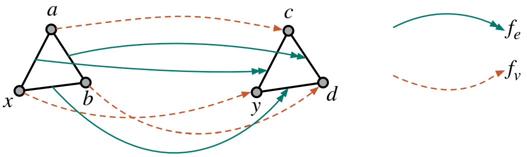

FIGURE 3: Fibrations map vertices to vertices and edges to edges. When three vertices form a triangle, fibrations also preserve the triangle structure. Therefore, one can define an isomorphism between local neighbourhoods using fibrations.

Our local symmetry-driven lumping approach shares a close relationship with what are known as fibrations in algebraic graph theory. We briefly describe the relationship in the following.

6. Graph fibrations

Fibrations of graphs were first inspired by fibrations between a pair of categories [9]. Although the idea of fibrations originated from category theory, it has deep implications for graph theory, theoretical computer science, and other mathematical disciplines. For instance, Boldi, Lonati, Santini, and Vigna [8] discussed its interesting connections to the PageRank citation ranking algorithm. Nijholt, Rink, and Sanders [33] explored the similarities between dynamical systems with a network structure and dynamical systems with symmetry by means of fibrations of graphs. Let us now define the necessary graph-theoretic concepts.

Given the graph $G = (V, E)$, we first define the source and target maps $\mathsf{s}_G, \mathsf{t}_G \colon E \longrightarrow V$ on G such that $\mathsf{s}_G(u, v) = u$ and $\mathsf{t}_G(u, v) = v$ for each $(u, v) \in E$. Let $H = (V', E')$ be another graph. The source and the target maps $\mathsf{s}_H, \mathsf{t}_H$ are defined analogously. A map $f := (f_v, f_e)$, where $f_v : V \longrightarrow V'$ and $f_e \colon E \longrightarrow E'$, is called a *graph morphism* between G and H (from G to H, to be precise) if f_v and f_e commute with the source and the target maps of G and H, i.e. if $\mathsf{s}_H f_e = f_v \mathsf{s}_G$ and $\mathsf{t}_H f_e = f_v \mathsf{t}_G$. A morphism is called an *epimorphism* if both f_v and f_e are surjective. Finally, we define a graph fibration as follows [9].

Definition 6. A morphism $f := (f_v, f_e)$ between two graphs $G = (V, E)$ and $H = (V', E')$ is called a fibration between graphs G and H (from G to H, to be precise) if, for each edge $a \in E'$ and for each $x \in V$ satisfying $f_v(x) = \mathsf{t}_H(a)$, there exists a unique edge $a_x \in E$ such that $f_e(a_x) = a$ and $\mathsf{t}_G(a_x) = x$. The edge a_x thus found is called the lifting of a at x, and is denoted by $f_e^{-1}(a)$. The graph G is then called fibred over H. The fibre over a vertex $y \in V'$, denoted by $\mathsf{fibre}(y)$, is the set of vertices in V that are mapped to y, i.e. $\mathsf{fibre}(y) := \{x \in V \mid f_v(x) = y\}$.

In the original paper [9], Boldi and Vigna defined colour-preserving graph morphisms when graphs are endowed with a colouring function. In that case, f_e also commutes with the colouring function. For our present purposes, we do not require this generality and only consider uncoloured graphs. Boldi and Vigna [9] showed that a left action of a group on G can be used to induce fibrations. They also showed that fibrations and epimorphisms satisfying certain local in-isomorphism property are equivalent [9, Theorem 2]. Indeed, fibrations have a close relationship with the notion of local symmetry described in Section 5. The proof of the following proposition follows analogously from [9, Theorem 2]. However, for the sake of completeness, we also provide it in Appendix A.

Proposition 4. *Let $f := (f_v, f_e)$ be a fibration of the graph $G = (V, E)$, i.e. a fibration from G to G itself. Pick two vertices $x, y \in V$. If $x \in \mathsf{fibre}(y)$, the vertices x, y are locally symmetric,*

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

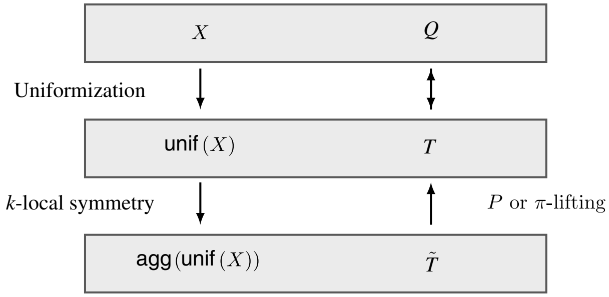

FIGURE 4: Lifting procedure used to assess the quality of the approximation.

i.e. $x\overset{1}{\sim}y$. *Moreover, if the vertices* x, y *are locally symmetric, there exists a fibration such that* $x \in \mathsf{fibre}(y)$.

The above proposition essentially shows that the equivalence classes of local symmetry (with $k=1$) and fibres induced by a graph fibration are the same. Therefore, the fibres can also be used to aggregate the states of $\mathcal{X}^N$ to achieve approximate lumpability in the same fashion as we did with local symmetry. See Figure 3.

7. Approximation error

As the lumping based on local symmetry does not ensure Markovianness of the lumped process, we need to quantify the approximation error. In order to do so, we work with the uniformization of X. Then we lump $\mathsf{unif}(X)$ to produce $\mathsf{agg}(\mathsf{unif}(X))$ according to k-local symmetry. A direct assessment of the quality of aggregation is cumbersome. Therefore, it is suggested [15, 20] that we *lift* the aggregated process $\mathsf{agg}(\mathsf{unif}(X))$ to a Markov chain on the same state space $\mathcal{X}^N$ as $\mathsf{unif}(X)$ and then compare their transition probability matrices. The lifting allows us to use known metrics of divergence such as the Kullback–Leibler divergence to quantify the approximation error. We follow the scheme depicted in Figure 4.

In order to fix ideas, let us lump $\mathsf{unif}(X)$ according to k-local symmetry, i.e. according to the partition $\{\mathcal{X}_1^{(k)}, \mathcal{X}_2^{(k)}, \dots, \mathcal{X}_{M_k}^{(k)}\}$ of $\mathcal{X}^N$ obtained as the equivalence classes of $\overset{k}{\sim}$. We introduce two notations in this connection. Let $\eta_k \colon \mathcal{X}^N \longrightarrow [M_k]$ be the partition function associated with $\overset{k}{\sim}$, i.e. $\eta_k(x) := i$ if and only if $x \in \mathcal{X}_i^{(k)}$. For $u \in \mathcal{X}^N$, let us denote the equivalence class containing u by $\langle x\rangle_k$, i.e. $\langle x\rangle_k := \mathcal{X}_i^{(k)}$ if and only if $x \in \mathcal{X}_i^{(k)}$. Note that $\langle x\rangle_k = \eta_k^{-1}(\eta_k(x))$.

Let $T = ((t_{i,j}))$ be the transition probability matrix associated with $\mathsf{unif}(X)$. Now, since X is not necessarily lumpable with respect to the partition $\{\mathcal{X}_1^{(k)}, \mathcal{X}_2^{(k)}, \dots, \mathcal{X}_{M_k}^{(k)}\}$, for $i \neq j \in [M_k]$ and two distinct $x, y \in \mathcal{X}_i^{(k)}$, the quantity $\sum_{z\in\mathcal{X}_j^{(k)}} t_{x,z}$ may not equal $\sum_{z\in\mathcal{X}_j^{(k)}} t_{y,z}$. In other words, $\mathsf{agg}(\mathsf{unif}(X))$ may not be Markovian. The idea is to approximate $\mathsf{agg}(\mathsf{unif}(X))$ with a Markov chain and then quantify the approximation error. Therefore, in order to avoid additional notations, we proceed as if $\mathsf{agg}(\mathsf{unif}(X))$ were Markovian and estimate its transition probabilities. If $\mathsf{unif}(X)$ is stationary with distribution π, i.e. if π is the solution to $\pi T = \pi$ and $p(0) = \pi$, a natural estimate of the transition probability of the lumped process is as follows:

$$\tilde{t}_{i,j}^{(k)} := \frac{\sum_{u\in\mathcal{X}_i^{(k)}} \pi_u \sum_{v\in\mathcal{X}_j^{(k)}} t_{u,v}}{\sum_{u\in\mathcal{X}_i^{(k)}} \pi_u} \qquad \text{for } i, j \in [M_k].$$

That is, we estimate the transition probabilities of the (Markov chain that approximates the) lumped process $\mathsf{agg}(\mathsf{unif}(X))$ by averaging the different values $\sum_{z\in\mathcal{X}_j^{(k)}} t_{x,z}$ and $\sum_{z\in\mathcal{X}_j^{(k)}} t_{y,z}$,

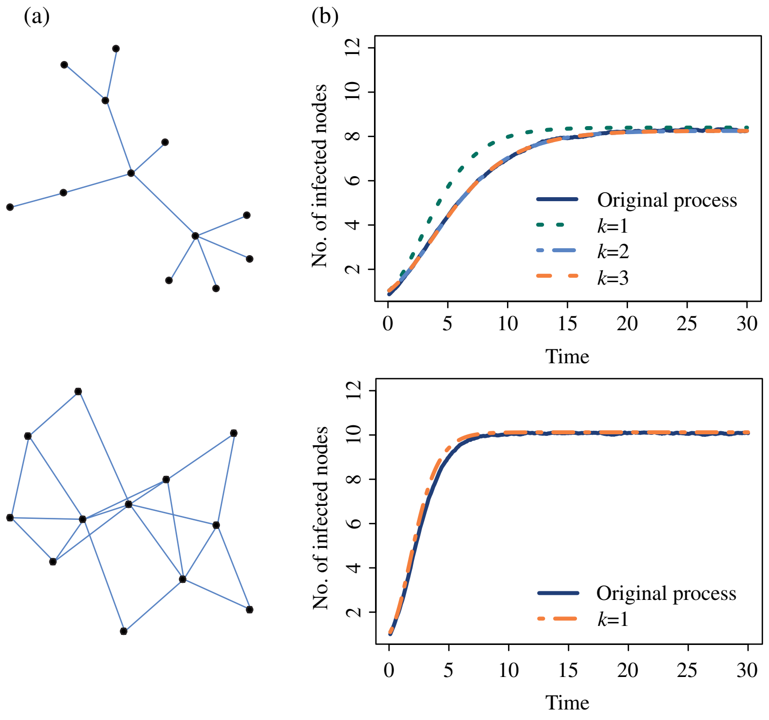

FIGURE 5: (a) Two graphs, and (b) comparisons of the expected number of infected nodes for SIS dynamics on the graphs.

weighted by the stationary probabilities [20]. Let $\tilde{T}^{(k)} := ((\tilde{t}_{i,j}^{(k)}))$. Before we proceed to explain the lifting procedure, let us consider an example.

Example 6. Consider SIS processes on the two graphs each with 12 vertices, shown in Figure 5. The graphs are shown in (a); in (b) we plot the expected number of infected nodes over time calculated from the reduced system for increasing k against the true expected number of infected nodes calculated from the original process. As one would expect, Figure 5 shows that there is close agreement between the true expected numbers and those obtained from the reduced systems even for small values of k.

Now, we describe how the transition probabilities of the lifted Markov chain are calculated. There are two common ways of lifting $\mathsf{agg}(\mathsf{unif}(X))$ to a Markov chain on $\mathcal{X}^N$, one using a probability vector, called π-lifting, and the other using the transition probabilities, called P-lifting. Let us discuss π-lifting first.

Definition 7. (π*-lifting.*) The π-lifting of $\eta_k(\mathsf{unif}(X))$ is a DTMC with transition probability matrix $T_k^{\pi} := ((t_{u,v}^{\pi,k}))$, where

$$t_{u,v}^{\pi,k} := \frac{\pi_v}{\sum_{x\in\langle v\rangle_k} \pi_x} \tilde{t}^{(k)}_{\eta_k(u),\eta_k(v)}, \qquad \text{where } u, v \in \mathcal{X}^N.$$

Note that, in principle, π-lifting can be done using any probability vector as long as the denominator remains non-zero for the choice of the candidate probability vector. Nevertheless, the most common choice is the stationary probability vector. The reason for this choice is the fact that the stationary probability vector achieves the minimum KL divergence rate [15].

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

For this reason, in this paper we consider π-lifting with the stationary distribution for numerical computations. Another immediate consequence of π-lifting is that the lifted Markov chain with transition probability matrix T_k^π given in (7) is lumpable with respect to the partition $\mathcal{X}^N/\overset{k}{\sim}$ and has π as the stationary probability. In many situations, computing the stationary distribution π may not be straightforward. Many numerical methods can be adopted in such situations. See Section 8 for a more detailed discussion. Now, we define the approximation error.

Definition 8. We define the approximation error to be the KL divergence rate between $\mathsf{unif}(X)$ and the lifted DTMCs. Therefore, for π-lifting, the approximation error is given by

$$D_{\mathrm{KL}}(T \mid\mid T_k^\pi) := \sum_{u\in\mathcal{X}^N}\sum_{v\in\mathcal{X}^N}\pi_u t_{u,v}\log\left(\frac{t_{u,v}}{t_{u,v}^{\pi,k}}\right).$$

Having defined the approximation error, we show that it indeed decreases monotonically with the order of local symmetry. This is precisely the assertion of Theorem 2. However, in order to prove Theorem 2, we need to make use of the following calculation, which we present in the form of a lemma.

Lemma 2. *For any two states $u, v \in \mathcal{X}^N$, and for any k, define the ratio*

$$\rho_k(u, v) := \frac{\sum_{t\in\langle v\rangle_k}\pi_t}{\tilde{t}^{(k)}_{\eta_k(u),\eta_k(v)}} = \frac{\sum_{p\in\langle u\rangle_k}\pi_p\sum_{q\in\langle v\rangle_k}\pi_q}{\sum_{p\in\langle u\rangle_k}\sum_{q\in\langle v\rangle_k}\pi_p t_{p,q}}.$$

Then, the following recursion relation holds:

$$\sum_{x\in\langle u\rangle_k}\sum_{y\in\langle v\rangle_k}\pi_x t_{x,y}\rho_{k+1}(x, y) = \rho_k(u, v)\sum_{x\in\langle u\rangle_k}\sum_{y\in\langle v\rangle_k}\pi_x t_{x,y}.$$

Proof of Lemma 2. By the refinement property of local symmetry in Proposition 3, we can find distinct integers $i_1, i_2, \ldots, i_m$ and $j_1, j_2, \ldots, j_n$ in $[M_{k+1}]$ such that

$$\langle u\rangle_k = \cup_{l\in[m]}\mathcal{X}^{(k+1)}_{i_l} \quad\text{and}\quad \langle v\rangle_k = \cup_{l\in[n]}\mathcal{X}^{(k+1)}_{j_l}. \tag{7.1}$$

Therefore, we can split the summation over $\langle u\rangle_k$, $\langle v\rangle_k$ into disjoint equivalence classes of $\overset{k+1}{\sim}$. Within each of these equivalence classes of $\overset{k+1}{\sim}$, the quantity $\tilde{t}_{\eta_{k+1}(x),\eta_{k+1}(y)}$ is constant and can therefore be pulled out of the summation. Therefore,

$$\begin{aligned}
&\sum_{x\in\langle u\rangle_k}\sum_{y\in\langle v\rangle_k}\pi_x t_{x,y}\rho_{k+1}(x, y)\\
&\quad= \sum_{p\in[m]}\sum_{q\in[n]}\sum_{x\in\mathcal{X}^{(k+1)}_{i_p}}\sum_{y\in\mathcal{X}^{(k+1)}_{j_q}}\pi_x t_{x,y}\left(\frac{\sum_{s\in\langle x\rangle_{k+1}}\pi_s\sum_{t\in\langle y\rangle_{k+1}}\pi_t}{\sum_{s\in\langle x\rangle_{k+1}}\sum_{t\in\langle y\rangle_{k+1}}\pi_s t_{s,t}}\right)\\
&\quad= \sum_{p\in[m]}\sum_{q\in[n]}\left(\frac{\sum_{s\in\mathcal{X}^{(k+1)}_{i_p}}\pi_s\sum_{t\in\mathcal{X}^{(k+1)}_{j_q}}\pi_t}{\sum_{s\in\mathcal{X}^{(k+1)}_{i_p}}\sum_{t\in\mathcal{X}^{(k+1)}_{j_q}}\pi_s t_{s,t}}\right)\sum_{x\in\mathcal{X}^{(k+1)}_{i_p}}\sum_{y\in\mathcal{X}^{(k+1)}_{j_q}}\pi_x t_{x,y}\\
&\quad= \sum_{x\in\langle u\rangle_k}\sum_{y\in\langle v\rangle_k}\pi_x\pi_y\\
&\quad= \rho_k(u, v)\sum_{x\in\langle u\rangle_k}\sum_{y\in\langle v\rangle_k}\pi_x t_{x,y}.
\end{aligned}$$

This completes the proof. □

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

Note that $\rho_k(u, v) = \rho_k(x, y)$ for any $u \overset{k}{\sim} x$ and $v \overset{k}{\sim} y$. Therefore, we can use the shorthand notation $\rho_k(\mathcal{X}_i^{(k)}, \mathcal{X}_j^{(k)})$ to mean $\rho_k(u, v)$ for any $u \in \mathcal{X}_i^{(k)}$, $v \in \mathcal{X}_j^{(k)}$.

Remark 5. (*Averaging argument.*) The main implication of Lemma 2 is that the quantity $\rho_k(u, v)$ can be seen as a weighted average of $\rho_{k+1}(x, y)$ where the x, y are in the equivalence classes of $\overset{k+1}{\sim}$. The weights are precisely

$$W_{\langle u\rangle_k, \langle v\rangle_k}(\mathcal{X}_{i_p}^{(k+1)}, \mathcal{X}_{j_q}^{(k+1)}) := \frac{\sum_{x\in\mathcal{X}_{i_p}^{(k+1)}} \sum_{y\in\mathcal{X}_{j_q}^{(k+1)}} \pi_x t_{x,y}}{\sum_{x\in\langle u\rangle_k} \sum_{y\in\langle v\rangle_k} \pi_x t_{x,y}},$$

where we have partitioned $\langle u\rangle_k$ and $\langle v\rangle_k$ into $\mathcal{X}_{i_p}^{(k+1)}$ and $\mathcal{X}_{j_q}^{(k+1)}$ respectively, as shown in (7.1). We interpret $W_{\langle u\rangle_k, \langle v\rangle_k}(\mathcal{X}_{i_p}^{(k+1)}, \mathcal{X}_{j_q}^{(k+1)})$ as the weight for the cross-section $\mathcal{X}_{i_p}^{(k+1)} \times \mathcal{X}_{j_q}^{(k+1)}$ with regard to the partition of $\langle u\rangle_k$ and $\langle y\rangle_k$ given in (7.1). Therefore, it follows from Lemma 2 that

$$\rho_k(\langle u\rangle_k, \langle v\rangle_k) = \sum_{p\in[m]} \sum_{q\in[n]} \rho_{k+1}(\mathcal{X}_{i_p}^{(k+1)}, \mathcal{X}_{j_q}^{(k+1)}) W_{\langle u\rangle_k, \langle v\rangle_k}(\mathcal{X}_{i_p}^{(k+1)}, \mathcal{X}_{j_q}^{(k+1)}).$$

Since the weights sum up to unity, $\rho_k(u, v)$ can indeed be seen as an average. Keeping this remark in mind, we now proceed to state and prove Theorem 2 about the monotonicity of the approximation error.

Theorem 2. *For π-lifting, the aggregation of states in $\mathcal{X}^N$ using local symmetry ensures monotonically decreasing approximation error with increasing order of local symmetry, that is,*

$$D_{KL}(T \,||\, T_{k+1}^{(\pi)}) \le D_{KL}(T \,||\, T_k^{(\pi)}) \quad \textit{for all } k \ge 1.$$

Proof of Theorem 2. By the refinement property of local symmetry proved in Proposition 3, we can partition $[M_{k+1}] = \{1, 2, \ldots, M_{k+1}\}$ into $\{\Lambda_1, \Lambda_2, \ldots, \Lambda_{M_k}\}$ such that

$$\mathcal{X}_i^{(k)} = \cup_{l\in\Lambda_i} \mathcal{X}_l^{k+1}.$$

Note that

$$\begin{aligned}
&D_{\mathrm{KL}}(T \,||\, T_k^{(\pi)}) - D_{\mathrm{KL}}(T \,||\, T_{k+1}^{(\pi)}) \\
&\quad = \sum_{i,j\in[M_k]} \sum_{u\in\mathcal{X}_i^{(k)}} \sum_{v\in\mathcal{X}_j^{(k)}} \pi_u t_{u,v} \log\left(\frac{\rho_k(u, v)}{\rho_{k+1}(u, v)}\right) \\
&\quad = \sum_{i,j\in[M_k]} (\log(\rho_k(\mathcal{X}_i^{(k)}, \mathcal{X}_j^{(k)})) \sum_{u\in\mathcal{X}_i^{(k)}} \sum_{v\in\mathcal{X}_j^{(k)}} \pi_u t_{u,v} - \sum_{u\in\mathcal{X}_i^{(k)}} \sum_{v\in\mathcal{X}_j^{(k)}} \pi_u t_{u,v} \log(\rho_{k+1}(u, v))) \\
&\quad = \sum_{i,j\in[M_k]} \Theta_{i,j},
\end{aligned}$$

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

where

$$\begin{aligned}\Theta_{i,j} :=\ & \log\left(\rho_k(\mathcal{X}_i^{(k)}, \mathcal{X}_j^{(k)})\right) \sum_{u\in\mathcal{X}_i^{(k)}} \sum_{v\in\mathcal{X}_j^{(k)}} \pi_u t_{u,v} \\ & - \sum_{p\in\Lambda_i} \sum_{q\in\Lambda_j} \sum_{u\in\mathcal{X}_p^{(k+1)}} \sum_{v\in\mathcal{X}_q^{(k+1)}} \pi_u t_{u,v} \log\left(\rho_{k+1}(u, v)\right) \\ =& \left(\sum_{u\in\mathcal{X}_i^{(k)}} \sum_{v\in\mathcal{X}_j^{(k)}} \pi_u t_{u,v} \right) \times \Bigg(\log\left(\rho_k(\mathcal{X}_i^{(k)}, \mathcal{X}_j^{(k)})\right) \\ & - \sum_{p\in\Lambda_i} \sum_{q\in\Lambda_j} W_{\mathcal{X}_i^{(k)},\mathcal{X}_j^{(k)}}(\mathcal{X}_p^{(k+1)}, \mathcal{X}_q^{(k+1)}) \log\left(\rho_{k+1}(\mathcal{X}_p^{(k+1)}, \mathcal{X}_q^{(k+1)})\right)\Bigg) \\ \geq\ & 0,\end{aligned}$$

by Jensen's inequality and the averaging argument given in Remark 5 and Lemma 2. This completes the proof. □

Note that $D_{\mathrm{KL}}(T \mid\mid T_k^{(\pi)}) - D_{\mathrm{KL}}(T \mid\mid T_{k+1}^{(\pi)}) = 0$ is achieved if (and only if) equality occurs in Jensen's inequality, forcing the individual $\Theta_{i,j}$ to be zeros. This is the case when the ρ_k and ρ_{k+1} are equal. There are two possibilities. First, the equivalence classes of $\overset{k}{\sim}$ and $\overset{k+1}{\sim}$ are the same. In this case, by Proposition 2, the equivalence classes of all $\overset{k+j}{\sim}$, for $j \geq 2$, will remain the same. Therefore, we have already reached automorphism, and hence exact lumpability. Second, the equivalence classes of $\overset{k}{\sim}$ and $\overset{k+1}{\sim}$ are different (so we are not yet at automorphism), but exact lumpability has already been achieved at the order of local symmetry k. In both cases, we need not refine our partition further because exact lumpability has been achieved. Therefore, $D_{\mathrm{KL}}(T \mid\mid T_k^{(\pi)}) - D_{\mathrm{KL}}(T \mid\mid T_{k+1}^{(\pi)}) = 0$ serves as a definite stopping rule for any iterative algorithmic implementation of local symmetry-driven lumping.

Now, we discuss the second type of lifting, which makes use of the transition probabilities and is called *P*-lifting. The following is the definition.

Definition 9. (*P-lifting.*) The *P*-lifting of $\eta_k(\mathsf{unif}(X))$ is a DTMC with transition probability matrix $T_k^P := ((t_{u,v}^{P,k}))$ where, for $u, v \in \mathcal{X}^N$,

$$t_{u,v}^{P,k} := \begin{cases} \dfrac{t_{u,v}}{\sum_{x\in\langle v\rangle_k} t_{u,x}} \tilde{t}^{(k)}_{\eta_k(u),\eta_k(v)} & \text{if } \sum_{x\in\langle v\rangle_k} t_{u,x} > 0, \\ \dfrac{1}{|\langle v\rangle_k|} \tilde{t}^{(k)}_{\eta_k(u),\eta_k(v)} & \text{if } \sum_{x\in\langle v\rangle_k} t_{u,x} = 0. \end{cases}$$

The approximation error for *P*-lifting is defined similarly. Note that *P*-lifting is sharp, in the sense that if the lumping is in fact exact, then $D_{\mathrm{KL}}(T \mid\mid T_k^{(P)}) = 0$, whereas π-lifting is not [20]. In Figure 6, we show numerical results pertaining to Theorem 2. We consider the Barabási–Albert preferential attachment, the Erdős–Rényi and the Watts–Strogatz small-world random graphs. The claimed monotonicity is observed in all three cases. In fact, the KL divergence rate decreases steeply in all three cases, for both π- and *P*-lifting. The figures are particularly encouraging in that a satisfactory level of accuracy is achieved even for small orders of local symmetry. Since one of the main purposes of aggregation is to engender state space reduction, we need to evaluate the performance of local symmetry-driven aggregation in terms of some

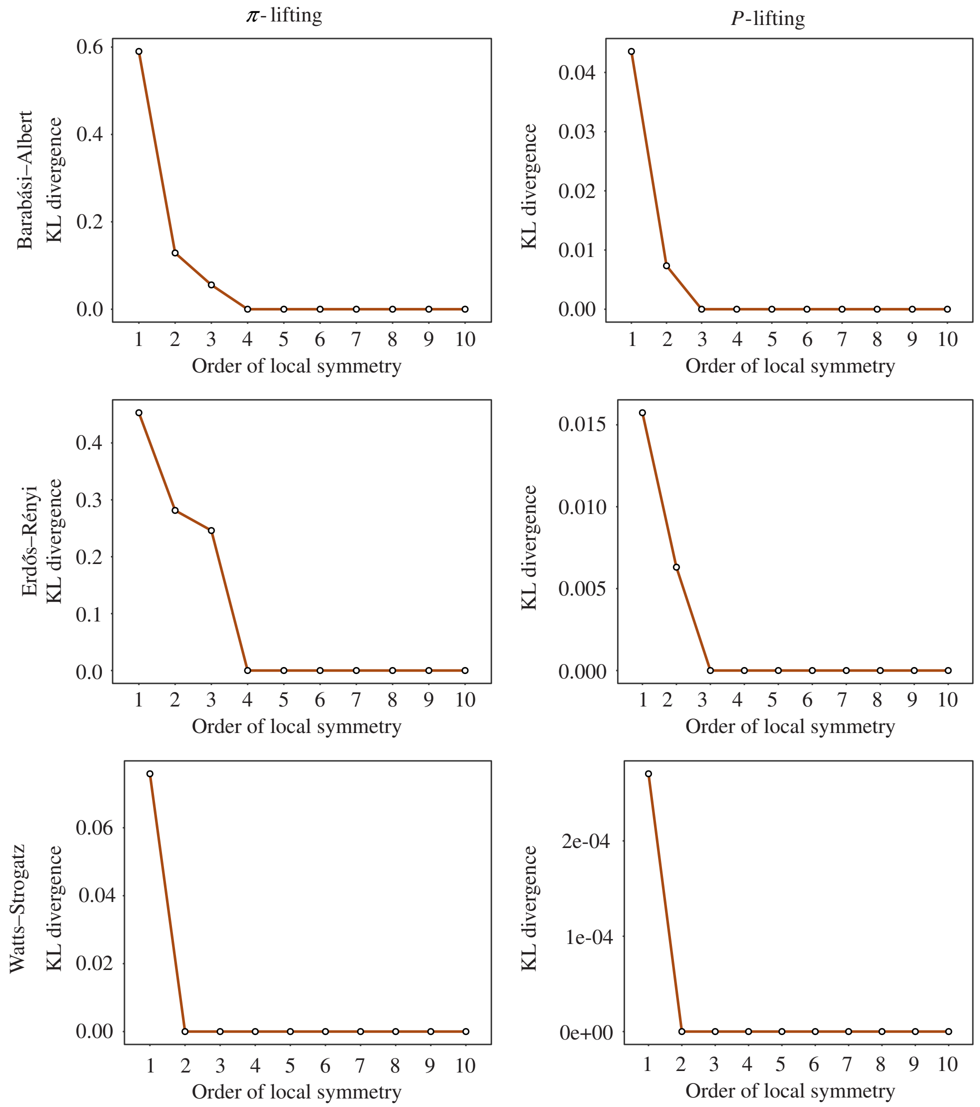

FIGURE 6: Monotonicity of the KL divergence with the order of the local symmetry for the SIS dynamics on different models of random graphs with 10 vertices. All graphs are undirected. The Erdős–Rényi graphs are created with edge probability 0.3, while the Watts–Strogatz small-world networks are created with rewiring probability 0.3 and each vertex having three neighbours. The infection and recovery rates are both 0.5.

notion of compression level as well. Therefore, we define compression level C at the order of local symmetry k as follows:

$$C(k) = 1 - \frac{M_k}{|\mathcal{X}^N|},$$

where M_k is the cardinality of the quotient space $\mathcal{X}^N/\overset{k}{\sim}$, i.e. the number of equivalence classes of $\overset{k}{\sim}$. If there is no non-trivial local symmetries, the compression level is zero

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

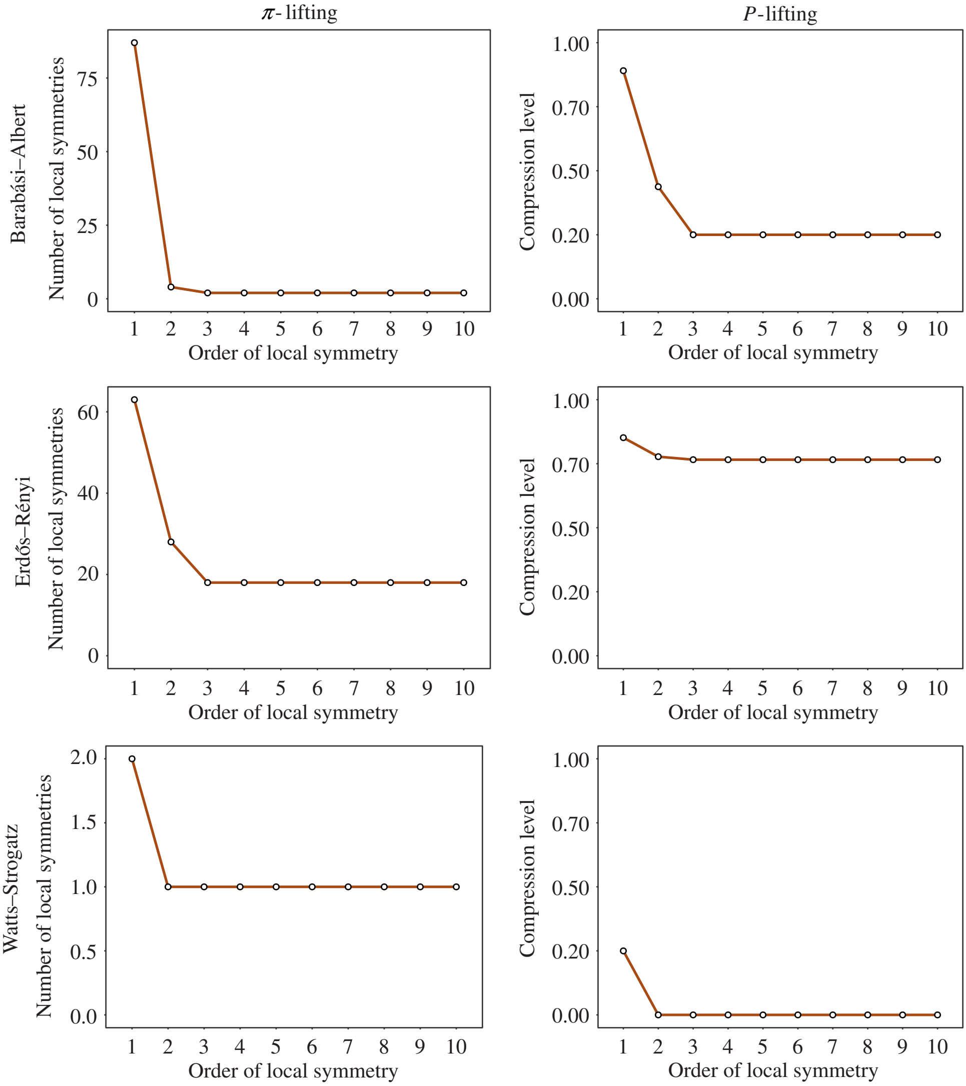

FIGURE 7: As we increase the order, the number of local symmetries (the cardinality of Ψ_k) decreases drastically. Therefore, the compression level also decreases. The simulation set-up is the same as in Figure 6.

because the partition is simply $\{\{x\} \mid x \in \mathcal{X}^N\}$. In Figure 7 we show how the number of local symmetries decreases drastically as we increase the order of local symmetry. Consequently, the compression level also falls steeply. This is expected because random graphs tend to become asymmetric as the number of vertices increases.

Remark 6. Note that Theorem 2 holds true for a general Markov chain whenever the partition function η_{k+1} is a refinement of η_k. The fact that the partition functions η_k, η_{k+1} are associated with the equivalence relations generated by k and $(k+1)$-local symmetries is only sufficient for the validity of Theorem 2, but not necessary. In fact, similar monotonicity can be proved, in a similar fashion, even when η_k, η_{k+1} are arbitrary partition functions defined on the state space

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

of a Markov chain such that η_{k+1} is a refinement of η_k. Notably, such monotonicity can only be guaranteed for π-lifting. In Appendix A, we provide a counter-example to establish that such monotonicity fails for P-lifting when arbitrary partition functions (one being a refinement of the other) are considered. However, this observation is about a general Markov chain. For our MABM, we observe similar monotonicity for P-lifting using numerical computations, as shown in Figure 6, but we cannot guarantee monotonicity in general.

8. Discussions

The idea of using Markov chain lumpability for model reduction has been discussed in the literature for some years now. For instance, Simon, Taylor, Kiss, and Miller [28, 39, 40] considered epidemiological scenarios, focusing mainly on binary dynamics. More general Markovian agent-based models were considered in [6]. Lumpability abstractions were applied to rule-based systems in [17] from a theoretical computer science perspective. While model reduction is one of the main purposes of lumpability, it is not the only one. In a recent paper [24], Katehakis and Smit identified a class of Markov chains, which they call successively lumpable and for which the stationary probabilities can be computed successively by computing stationary probabilities of a cleverly constructed sequence of Markov chains (typically on much smaller state spaces).

8.1. Coverings and colour refinements

For undirected graphs, a notion similar to our notion of local symmetry is called a *covering* [1]. However, in general, finding coverings is computationally challenging [29]. In our formulation, undirected graphs are to be treated as directed graphs with an edge set E satisfying $(i, j) \in E$ if and only if $(j, i) \in E$. The second notion that is similar to our approach is that of colour refinement [3, 7]. In order to draw an analogy, we think of the local states as colours, i.e. we have a K-colouring of G, and require isomorphisms to be colour-preserving. The objective is to devise a colouring method (given the initial colouring) that is *stable* in that two vertices with the same colour have identically coloured neighbourhoods. Note that a colouring naturally induces an equivalence relation on V. With successive refinement of colouring, we can construct equitable partitions of V in much the same way we did with local symmetry. The equitable partitions, in turn, can be used to yield approximately lumpable partitions of $\mathcal{X}^N$. Colour refinements are convenient and are often used as a simple isomorphism check. However, a limitation of this approach is that colour refinements are insufficient to find local isomorphisms in certain graphs such as regular graphs. In general, a graph G is said to be amenable to colour refinement if it is distinguishable from any other graph H via colour refinement. A number of classes of graphs are known to be amenable [3], e.g. unigraphs, trees. It is also known [5] that Erdős–Rényi random graphs are amenable with high probability.

8.2. Regular graphs

Large regular graphs, in general, can exhibit different dynamics. Since the vertices have similar neighbourhoods, our local symmetry will not be able to distinguish between them. This may lead to poor lumping. Increasing the order of local symmetry will avoid such issues. A theoretical analysis of this special case of regular graphs is planned for future work.

8.3. Computation of the stationary distribution

Note that computation of the stationary distribution itself is cumbersome for Markov chains on large state spaces. In many cases, the transition matrix is sparse, which makes available a

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

host of numerical tools developed for sparse matrices. There are also numerical algorithms [41], such as the Courtois method [14] or Takahasi's iterative aggregation–disaggregation method [42], for computing the stationary distribution. In general, the efficiency of Takahashi's algorithm depends on a good initial clustering of states. In our case, the computation is facilitated by the fact that the initial quotient space $\mathcal{X}^N/\overset{1}{\sim}$ is expectedly a better partition than a random one. In a recent paper [30], Kuntz, Thomas, Stan, and Barahona derived bounds on the stationary distribution (and moments) based on mathematical programming. In particular, when the stationary distribution is unique, they provide computable error bounds. Sampling-based techniques can also be used for this purpose. For instance, Hemberg and Barahona [22] provided an algorithm that combine Gillespie's algorithm with Dominated Coupling From The Past (DCFTP) techniques to provide guaranteed sampling from the stationary distribution.

8.4. Markov chain enlargement

An interesting concept closely related to aggregation is Markov chain enlargement. There are many examples where enlargement of the state space of a Markov chain can be computationally beneficial in that it can significantly reduce the mixing time. See [2, 13] for a discussion of how splitting up states of a Markov chain can speed up mixing. This has implications for the performance of statistical inference algorithms that rely on the mixing of some Markov chain, and also for optimization algorithms such as the alternating direction method of multipliers (ADMM). França and Bento [18] showed that, for certain objective functions, the distributed ADMM algorithm can indeed be seen as a lifting of the gradient descent algorithm.

8.5. CTBNs and SANs

The Markovian agent-based model that we consider in this paper belongs to a more general class of models known as interacting particle systems (IPS) in the probability literature. The tools developed here are expected to find applications beyond what has been discussed here, and are immediately applicable to many of the traditional IPS models arising from statistical physics, population biology, and social sciences. Such models include contact processes, voter models, and exclusion models. The MABM model discussed in the present paper is also closely related to continuous-time Bayesian networks (CTBNs) [34] and stochastic automata networks (SANs) [11]. To be specific, the local intensity functions defined in (3.1) constitute the conditional intensity matrix (CIM) in [34]. These CIMs can be then combined into Q by the 'amalgamation' operation. Another approach that is popular in the SAN literature is to give Q a Kronecker representation [11]. We expect that the present endeavour will help to bridge the gap between the various communities that make use of agent-based models.

Appendix A.

Proof of Proposition 1. It can be verified that $\{\tilde{\mathcal{Y}}_1, \tilde{\mathcal{Y}}_2, \ldots, \tilde{\mathcal{Y}}_M\}$ indeed forms a partition of $\mathcal{Y}$. Let us denote the transition rate matrix of Z by $\tilde{A} = ((\tilde{a}_{i,j}))$, where $\tilde{a}_{i,j} = a_{f^{-1}(i), f^{-1}(j)}$, and f^{-1} is the inverse of f in $\mathsf{Sym}(\mathcal{Y})$. The proof will be complete if we show that the linear system $\dot{z} = z\tilde{A}$ is lumpable. Pick $\tilde{\mathcal{Y}}_i$, and $\tilde{\mathcal{Y}}_j$ for $i \neq j$, and let $u, v \in \tilde{\mathcal{Y}}_i$ be arbitrarily chosen. See that $u \in \tilde{\mathcal{Y}}_i$ implies $f^{-1}(u) \in \mathcal{Y}_i$. Then,

$$\sum_{l \in \tilde{\mathcal{Y}}_j} \tilde{a}_{u,l} = \sum_{l \in \tilde{\mathcal{Y}}_j} a_{f^{-1}(u), f^{-1}(l)} = \sum_{l \in \mathcal{X}_j} a_{s,l} = \sum_{l \in \mathcal{X}_j} a_{t,l} = \sum_{l \in \tilde{\mathcal{Y}}_j} a_{f^{-1}(v), f^{-1}(l)} = \sum_{l \in \tilde{\mathcal{Y}}_j} \tilde{a}_{v,l},$$

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

where $s = f^{-1}(u)$, $t = f^{-1}(v) \in \mathcal{X}_i$ and the equality for s and t holds by virtue of the lumpability of Y. This verifies Dynkin's criterion for $\dot{z} = z\tilde{A}$. □

Proof of Proposition 4. Let us first assume $x \in \mathsf{fibre}(y)$. In order to prove the vertices x, y are locally symmetric, we construct an isomorphism $g\colon N_1(x) \longrightarrow N_1(y)$ between $G[N_1(x)]$ and $G[N_1(y)]$ as follows:

$$g(a) := \mathsf{s}_G f_e^{-1}(a, x), \qquad \text{for all } a \in N_1(x).$$

Indeed, $f_e^{-1}(a, x)$ is an edge in $G[N_1(y)]$, and therefore, $g(a) \in N_1(y)$. In order to check whether g is indeed an isomorphism, pick two vertices $a, b \in N_1(x)$ such that $(a, b) \in E$. If $b = x$, the assertion follows straightforwardly. Therefore, we consider $b \neq x$. Then, $(a, b) \in E$ implies the vertices a, b, and x form a triangle (see Figure 3).

Since f is a fibration, (f_v, f_e^{-1}) is also a morphism because f_v and f_e^{-1} also commute with the source and target maps of G, i.e. $\mathsf{s}_G f_e^{-1} = f_v \mathsf{s}_G$ and $\mathsf{t}_G f_e^{-1} = f_v \mathsf{t}_G$. Now, let us consider the edge (a, b) in $G[N_1(x)]$. Since f is a fibration, there exists a unique edge $f_e^{-1}(a, b) = (c, d) \in E$ such that $f_e(c, d) = (a, b)$, where $d \in \mathsf{fibre}(b)$. Then,

$$(c, d) = (\mathsf{s}_G f_e^{-1}(a, b), \mathsf{t}_G f_e^{-1}(a, b)) = (f_v(a), f_v(b)) = (\mathsf{s}_G f_e^{-1}(a, x), \mathsf{t}_G f_e^{-1}(b, x)).$$

Therefore, g is indeed an isomorphism between $G[N_1(x)]$ and $G[N_1(y)]$ proving $x \overset{1}{\sim} y$.

Now, we prove the second part of the proposition. Let us assume $x \overset{1}{\sim} y$. In order to define a fibration $f = (f_v, f_e)$, let us first pick representatives for the equivalence classes of $\overset{1}{\sim}$. Let the injective map $\mathsf{r}\colon V \longrightarrow V$ define the representatives, that is, for each $x \in V$, we have $\langle x \rangle_1 = \langle \mathsf{r}(x) \rangle_1$. Then, consider the following maps:

$$f_v(x) := \mathsf{r}(x), \quad \text{for all } x \in V, \quad \text{and} \quad f_e(a, b) = (g(a), g(b)),$$

where $g \in \Psi_1$ is such that $g(b) = \mathsf{r}(b)$. Note that the choice of g depends on (a, b). The epimorphism f defined above is indeed a fibration [9]. This concludes the proof. □

Monotonicity fails for P-lifting

It is intuitive that the monotonic decrease of KL divergence for finer partitions should carry over to lifting by the transition matrix. However, this is not the case, as the following counterexample shows. Consider a transition probability matrix:

$$T = \begin{pmatrix} 0.10 & 0.10 & 0.07 & 0.16 & 0.13 & 0.20 & 0.04 & 0.20 \\ 0.11 & 0.17 & 0.10 & 0.15 & 0.12 & 0.13 & 0.14 & 0.08 \\ 0.07 & 0.13 & 0.10 & 0.14 & 0.09 & 0.02 & 0.41 & 0.04 \\ 0.16 & 0.08 & 0.02 & 0.17 & 0.05 & 0.23 & 0.06 & 0.23 \\ 0.07 & 0.12 & 0.20 & 0.17 & 0.22 & 0.21 & 0.01 & 0.00 \\ 0.07 & 0.15 & 0.25 & 0.10 & 0.18 & 0.03 & 0.21 & 0.01 \\ 0.14 & 0.07 & 0.20 & 0.14 & 0.10 & 0.10 & 0.07 & 0.18 \\ 0.10 & 0.19 & 0.07 & 0.22 & 0.11 & 0.03 & 0.14 & 0.14 \end{pmatrix}.$$

Now consider two partitions:

$$\{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}\} \quad \text{and} \quad \{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\},$$

Clearly the latter partition is a refinement of the first. However, when we use P-lifting, the first partition yields a KL divergence of 0.0019067, while the second partition yields a higher KL divergence of 0.0308801.

Acknowledgements

The work was conducted when WKB was a graduate student and AA was a summer intern at the Technische Universität Darmstadt in Germany. WKB was supported by the German Research Foundation (DFG) as part of project C3 within the Collaborative Research Center (CRC) 1053 – MAKI. YD was supported by the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School CE at Technische Universität Darmstadt. We sincerely thank the anonymous reviewers whose constructive feedback significantly improved the quality of the paper.

References

[1] Angluin, D. (1980). Local and global properties in networks of processors (extended abstract). In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing (STOC '80)*, pp. 82–93. ACM.

[2] Apers, S., Ticozzi, F. and Sarlette, A. (2017). Lifting Markov chains to mix faster: limits and opportunities. Available at arXiv:1705.08253.

[3] Arvind, V., Köbler, J., Rattan, G. and Verbitsky, O. (2016). Graph isomorphism, color refinement, and compactness. *Comput. Complexity* **26**, 627–685.

[4] Babai, L. (2016). Graph isomorphism in quasipolynomial time (extended abstract). In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing (STOC '16)*, pp. 684–697. ACM.

[5] Babai, L., Erdős, P. and Selkow, S. M. (1980). Random graph isomorphism. *SIAM J. Comput.* **9**, 628–635.

[6] Banisch, S. (2016). *Markov Chain Aggregation for Agent-Based Models*. Springer.

[7] Berkholz, C., Bonsma, P. and Grohe, M. (2013). Tight lower and upper bounds for the complexity of canonical colour refinement. In *Proceedings of the 21st Annual European Symposium on Algorithms (ESA 2013)* (Lecture Notes in Computer Science **8125**), pp. 145–156. Springer.

[8] Boldi, P., Lonati, V., Santini, M. and Vigna, S. (2006). Graph fibrations, graph isomorphism, and PageRank. *RAIRO Theoret. Inform. Appl.* **40**, 227–253.

[9] Boldi, P. and Vigna, S. (2002). Fibrations of graphs. *Discrete Math.* **243**, 21–66.

[10] Buchholz, P. (1994). Exact and ordinary lumpability in finite Markov chains. *J. Appl. Prob.* **31**, 59–75.

[11] Buchholz, P. and Kemper, P. (2004). Kronecker based matrix representations for large Markov models. In *Validation of Stochastic Systems* (Lecture Notes in Computer Science **2925**), pp. 256–295. Springer.

[12] Chatterjee, S. and Durrett, R. (2009). Contact processes on random graphs with power law degree distributions have critical value 0. *Ann. Prob.* **37**, 2332–2356.

[13] Chen, F., Lovász, L. and Pak, I. (1999). Lifting Markov chains to speed up mixing. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing (STOC '99)*, pp. 275–281. ACM, New York.

[14] Courtois, P. J. (1977). *Decomposability: Queueing and Computer System Applications*. Academic Press, New York.

[15] Deng, K., Mehta, P. G. and Meyn, S. P. (2011). Optimal Kullback–Leibler aggregation via spectral theory of Markov chains. *IEEE Trans. Automat. Control* **56**, 2793–2808.

[16] Elbert Simões, J., Figueiredo, D. R. and Barbosa, V. C. (2016). Local symmetry in random graphs. Available at arXiv:1605.01758.

[17] Feret, J., Henzinger, T., Koeppl, H. and Petrov, T. (2012). Lumpability abstractions of rule-based systems. *Theoret. Comput. Sci.* **431**, 137–164.

[18] França, G. and Bento, J. (2017). Markov chain lifting and distributed ADMM. *IEEE Signal Process. Lett.* **24**, 294–298.

[19] Ganguly, A., Petrov, T. and Koeppl, H. (2014). Markov chain aggregation and its applications to combinatorial reaction networks. *J. Math. Biol.* **69**, 767–797.

[20] Geiger, B. C., Petrov, T., Kubin, G. and Koeppl, H. (2015). Optimal Kullback–Leibler aggregation via information bottleneck. *IEEE Trans. Automat. Control* **60**, 1010–1022.

[21] Godsil, C. and Royle, G. F. (2013). *Algebraic Graph Theory* (Graduate Texts in Mathematics **207**). Springer, New York.

[22] Hemberg, M. and Barahona, M. (2008). A Dominated Coupling From The Past algorithm for the stochastic simulation of networks of biochemical reactions. *BMC Systems Biology* **2**, 42.

[23] Hendrickx, J. M. (2014). Views in a graph: to which depth must equality be checked? *IEEE Trans. Parallel Distrib. Systems* **25**, 1907–1912.

[24] Katehakis, M. N. and Smit, L. C. (2012). A successive lumping procedure for a class of Markov chains. *Probab. Engrg Inform. Sci.* **26**, 483–508.

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press

[25] Kemeny, J. G. and Snell, J. L. (1960). *Finite Markov Chains*. Van Nostrand, Princeton, NJ.

[26] KhudaBukhsh, W. R., Rückert, J., Wulfheide, J., Hausheer, D. and Koeppl, H. (2016). Analysing and leveraging client heterogeneity in swarming-based live streaming. In *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, pp. 386–394. IEEE.

[27] Kim, J. H., Sudakov, B. and Vu, V. H. (2002). On the asymmetry of random regular graphs and random graphs. *Random Structures Algorithms* **21**, 216–224.

[28] Kiss, I. Z., Miller, J. C. and Simon, P. L. (2017). *Mathematics of Epidemics on Networks: From Exact to Approximate Models* (Interdisciplinary Applied Mathematics **46**). Springer.

[29] Kratochvíl, J., Proskurowski, A. and Telle, J. A. (1998). Complexity of graph covering problems. *Nordic J. Comput.* **5**, 173–195.

[30] Kuntz, J., Thomas, P., Stan, G.-B. and Barahona, M. (2017). Rigorous bounds on the stationary distributions of the chemical master equation via mathematical programming. Available at arXiv:1702.05468.

[31] Łuczak, T. (1988). The automorphism group of random graphs with a given number of edges. *Math. Proc. Cambridge Phil. Soc.* **104**, 441–449.

[32] McKay, B. D. and Wormald, N. C. (1984). Automorphisms of random graphs with specified vertices. *Combinatorica* **4**, 325–338.

[33] Nijholt, E., Rink, B. and Sanders, J. (2016). Graph fibrations and symmetries of network dynamics. *J. Diff. Equations* **261**, 4861–4896.

[34] Nodelman, U., Shelton, C. R. and Koller, D. (2002). Continuous time Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pp. 378–387. Morgan Kaufmann.

[35] Norris, N. (1995). Universal covers of graphs: isomorphism to depth $n-1$ implies isomorphism to all depths. *Discrete Appl. Math.* **56**, 61–74.

[36] Riordan, O. and Wormald, N. (2010). The diameter of sparse random graphs. *Combin. Probab. Comput.* **19**, 835–926.

[37] Rubino, G. and Sericola, B. (1989). On weak lumpability in Markov chains. *J. Appl. Prob.* **26**, 446–457.

[38] Rubino, G. and Sericola, B. (1993). A finite characterization of weak lumpable Markov processes, II: The continuous time case. *Stochastic Process. Appl.* **45**, 115–125.

[39] Simon, P. L. and Kiss, I. Z. (2012). From exact stochastic to mean-field ODE models: a new approach to prove convergence results. *IMA J. Appl. Math.* **78**, 945–964.

[40] Simon, P. L., Taylor, M. and Kiss, I. Z. (2011). Exact epidemic models on graphs using graph-automorphism driven lumping. *J. Math. Biol.* **62**, 479–508.

[41] Stewart, W. J. (2000). Numerical methods for computing stationary distributions of finite irreducible Markov chains. In *Computational Probability* (International Series in Operations Research & Management Science **24**), pp. 81–111. Springer.

[42] Takahashi, Y. (1975). A lumping method for numerical calculations of stationary distributions of Markov chains. B-18, Department of Information Sciences, Tokyo Institute of Technology.

[43] Yamashita, M. and Kameda, T. (1996). Computing on anonymous networks, I: Characterizing the solvable cases. *IEEE Trans. Parallel Distrib. Systems* **7**, 69–89.

https://doi.org/10.1017/jpr.2019.44 Published online by Cambridge University Press