Compositional Approximate Markov Chain Aggregation for PEPA Models

Compositional Approximate Markov Chain Aggregation for PEPA Models

Dimitrios Milios and Stephen Gilmore

School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB, UK

Abstract. Approximate Markov chain aggregation involves the construction of a smaller Markov chain that approximates the behaviour of a given chain. We discuss two different approaches to obtain a nearly optimal partition of the state-space, based on different notions of approximate state equivalence.
Both approximate aggregation methods require an explicit representation of the transition matrix, a fact that renders them inefficient for large models. The main objective of this work is to investigate the possibility of compositionally applying such an approximate aggregation technique. We make use of the Kronecker representation of PEPA models, in order to aggregate the state-space of components rather than of the entire model.

1 Introduction

Markov chains have been used for many years for exploring the dynamic properties of systems that exhibit stochastic behaviour. They are supported by a great variety of techniques to obtain the steady-state and the transient probability distributions of such models. Many modelling formalisms generate Markov chains given some high-level description of the system. Unfortunately, even apparently simple models can generate extremely large state-spaces, a problem known as *state-space explosion*.

State-space aggregation can be an effective way to reduce the complexity of large Markov models. Aggregated models feature a reduced number of states, a fact that can accelerate transient and steady-state analysis techniques. Aggregation can be either exact or approximate. Exact aggregation of a Markov chain involves constructing a model with a smaller number of states that exhibits behaviour identical to that of the original system. If the original model is *lumpable*, then the resulting aggregated model will be a Markov chain as well. In the case of non-lumpable models, we use a reduced Markov model that approximates the behaviour of the original system. In this way, the model can be solved efficiently at the cost of loss of accuracy.

Existing approximate aggregation techniques [7][23][6] typically require the computation of several eigenvectors of the probability matrix. The great computational cost of this requirement renders these approaches not particularly popular in performance modelling. Instead, we take advantage of a compositional

representation of the state-space, in order to apply approximate aggregation techniques on components rather than the entire system. The resulting reduced components are then combined to form an overall reduced state-space.

The modelling paradigm which we work with here is the PEPA language [13], and its Kronecker representation [15] in particular. A PEPA model is represented as a collection of interacting components, and each one of these has its own state-space and performs a number of actions that change its internal state. The global state of the system is expressed in terms of the local states of the components included. PEPA components can be considered as labelled continuous-time Markov chains (CTMC). We reduce the state-space of more than one component at the same time. Intuitively, the more components we approximate, the greater reduction of the global state-space we can achieve.

In order to partition the state-space of these CTMCs we apply two different approaches. The first one is a traditional approach which is related to near complete decomposability (NCD) and the spectral properties of Markov chains. The second one is a novel method that relies on the notion of quasi-lumpability. We note that in most cases we make use of the embedded discrete time Markov chain that is obtained after *uniformisation* [16].

Related work is outlined in Sect. 2. Section 3 briefly outlines the NCD-based approach. In Sect. 4 we present our approximate aggregation approach that is based on quasi-lumpability. In Sect. 5 we describe how approximate aggregation is applied in a compositional setting. Section 6 involves examples that demonstrate the performance of the two aggregation techniques. Finally, the conclusions and considerations for future work are summarized in Sect. 7.

2 Related Work

In terms of Markov chains, equivalence is formally described by the notion of *lumpability* [17]. As can be seen in [1], given a *lumpable* Markov chain we can obtain a *lumped* model which is also a Markov chain having identical transient and steady-state behaviour. State-space aggregation techniques that rely on this concept typically exploit the structure of some high-level description of the model. For example in [11], a lumpable partition is obtained by identifying isomorphic components of a PEPA model. In the general case though, a lumpable partition might not exist.

Quasi-lumpability, which was introduced in [9], captures approximate behaviour for Markov models. The term *near-lumpability* has been used to describe the same notion in [1], where the concept was generalized towards exactly and strictly lumpable Markov chains. Since we are interested in *nearly ordinarily lumpable* Markov models only, we shall use the term quasi-lumpability for the rest of the paper. Most of the research in the field so far aims at computing bounds for the state probabilities of quasi-lumpable Markov chains, assuming some partition of the state-space [9][10][8][2]. The computation of bounds of compositions of Markov chains has also been investigated in the context of Markov reward models [4] and PEPA [24]. Our goal is to develop a strategy to

automatically obtain a partition of the state-space that is nearly optimal with respect to a measure related to quasi-lumpability.

Many existing approximate Markov chain aggregation techniques ([7][6]) rely on the notion of *near complete decomposability* (NCD) [3]. By definition, a completely decomposable Markov chain consists of uncoupled aggregates of states, which means that a random walk will never transition from one aggregate to another. This restriction is relaxed for nearly completely decomposable systems, where the aggregates are almost uncoupled. The relation between the spectral properties of probability matrices and NCD has been investigated in a number of works [20][22][12]. In [7], the structure of the eigenvectors has been used to partition the state-space of reversible Markov chains, in a way that minimizes the probability of transitioning between partitions. In [23], this framework has been extended to non-reversible models. In a more recent work [6], a similar approach for partitioning Markov models has been presented which is based on information theory.

However, spectral methods are not directly related to the notion of lumpability which formally captures equivalence between Markov chains. At this point, it is important to make a clear distinction between nearly completely decomposable and quasi-lumpable models. A Markov chain is nearly completely decomposable when there is a very small probability of transitioning from one part of the system to another, a fact that also implies quasi-lumpability as shown in [5]. In the general case however, a lumpable or a quasi-lumpable model does not have to be nearly completely decomposable. In this paper, we present results with respect to both quasi-lumpability and NCD approaches to approximately aggregate Markov models.

3 Aggregation based on NCD

3.1 Spectral Segmentation of Markov Chains

Let us consider a reversible Markov chain with probability matrix P and steady-state distribution $\boldsymbol{\pi}$. If $\Delta = \{A_1, \ldots, A_k\}$ is a partition of the state-space, we define the probability of the system moving from A_i to A_j in a single step:

$$Pr(A_i, A_j) = \frac{\sum_{i \in A_i, j \in A_j} \pi_i P_{ij}}{\sum_{i \in A_i} \pi_i} \tag{1}$$

Given a completely decomposable Markov model, we have $Pr(A_i, A_i) = 1$ and $Pr(A_i, A_j) = 0, \forall i \neq j$. This means that if the system is within a set of states A_i, it will never transition out of A_i. This condition is relaxed for nearly completely decomposable systems, where there is only a small probability of transitioning between parts of the system.

The eigenstructure of a probability matrix contains information about which parts of the Markov chain are almost invariant. As can be seen in [7], a probability matrix P with K invariant aggregates of states will have K eigenvalues that are equal to 1. It has been shown that states that belong to the same invariant

set A_i have the same sign-structure when mapped onto the eigenvector that corresponds to eigenvalue $\lambda = 1$. Perturbation analysis that was performed in [7] shows that this property is mostly preserved for the largest K eigenvectors for a nearly completely decomposable system as well. Hence, the sign-structure of the corresponding eigenvectors has been used to identify almost invariant aggregates of states.

3.2 The Non-Reversible Case

One key assumption made in the previous section is that we have a reversible Markov chain. In order to apply spectral segmentation to non-reversible Markov chains, we have to construct a reversible chain that approximates the original. Given some Markov process with probability matrix P and steady-state probability vector $\boldsymbol{\pi}$, its time reversal will have transition matrix $\bar{P}$ with elements:

$$\bar{P}_{ij} = P_{ji}\frac{\pi_j}{\pi_i} \tag{2}$$

Of course in the reversible case, $P = \bar{P}$. In order to handle non-reversible models, we could construct a reversible one that shares some properties of the initial non-reversible Markov model and its time reversal. For instance, we consider the following process:

$$\tilde{P} = \frac{P + \bar{P}}{2} \tag{3}$$

In the equation above, $\tilde{P}$ can be thought of as the mean process of the two. It is trivial to show that $\tilde{P}$ is a stochastic matrix with steady-state distribution $\boldsymbol{\pi}$. A similar approach appeared in [23], where a so-called *multiplicative reversibilisation* $\tilde{P} = P\bar{P}$ has been applied instead. In both cases though, there is an implicit assumption that the original non-reversible model has properties similar to those of the corresponding reversible process. This is true up to some extent, as both models have the same steady-state distribution. Thus, the eigenstructure of $\tilde{P}$ is used to obtain a partition of the state-space of P. Equation (3) implies that the closer to reversible P is, the better the approximation of its eigenproperties will be, when using $\tilde{P}$. However in cases where this is not true, this assumption could be a significant source of error. This is a consideration we try to investigate experimentally in Sect. 6.

4 Aggregation based on Quasi-lumpability

4.1 A Pseudo-metric related to Quasi-Lumpability

Given a partitioning of the state-space, lumpability implies that states that belong to the same class have identical transition probabilities to each of the partitions. To describe states with approximately similar rather than identical behaviour, we have to relax this condition. Approximately similar behaviour is captured by the concept of *quasi-lumpability* [9]:

Definition 1 (Quasi-Lumpability). *A Markov chain with probability matrix P will be quasi-lumpable w.r.t. a partition $\Delta = \{A_1, \ldots, A_K\}$ with K equivalence classes, if for any two classes $A_k, A_l \in \Delta$, and for any two states $i, j \in A_k$:*

$$\left| \sum_{m \in A_l} P_{im} - \sum_{m \in A_l} P_{jm} \right| \leq \epsilon, \quad \epsilon \geq 0 \tag{4}$$

The quantity ϵ in the equation above corresponds to the maximum difference between elements that are assigned to the same class. If we consider the transition probability matrix P of a quasi-lumpable model, this can be represented as $P = P^{-} + P^{\epsilon}$, where P^{-} is a lumpable Markov chain and P^{ϵ} a matrix with no element greater than the ϵ quantity of (4). In general, most of the values of P^{ϵ} should be zero, while the non-zero elements should be small. As noted in [1], if ϵ is sufficiently small, the lumpable model with transition matrix P^{-} approximates the behaviour of the quasi-lumpable one.

Using (4), we can define a pseudo-metric that captures a kind of similarity distance between states. If we consider all the equivalence classes $A_1, \ldots, A_K$, we define the following quantity for any two states i, j that belong to the same equivalence class:

$$E_{i,j} = \sum_{l=1}^{K} \left| \sum_{m \in A_l} P_{im} - P_{jm} \right| \tag{5}$$

In the equation above, $E_{i,j}$ will be equal to zero, iff the Markov chain is lumpable with respect to the partition $\Delta = \{A_1, \ldots, A_K\}$. Since it is possible that $E_{i,j} = 0$ when $i \neq j$, $E_{i,j}$ is characterized as a pseudo-metric, rather than as a metric.

Hence, the optimal quasi-lumpable partition will be the one that minimizes the quantity $E_{i,j}$ for any two states in the same class. However, the value of $E_{i,j}$ depends not only on the transition probabilities of states i and j, but also on the way that the states are distributed across the classes. In other words, a different partitioning of the state-space will result in a completely different $E_{i,j}$ quantity for the very same i and j states. Thus, it is very difficult to design an algorithm that minimizes $E_{i,j}$ with respect to the partitioning.

Instead, we show that the pseudo-metric $E_{i,j}$ is bounded by a proper distance metric independent of the partitioning. Starting from (5), if we pull the inner sum out of the absolute value, we will have a larger value:

$$E_{i,j} \leq \sum_{l=1}^{K} \sum_{m \in A_l} |P_{im} - P_{jm}| \tag{6}$$

It is evident that the sums in the inequality above cover the entire state-space of the original Markov model. Thus, given that the initial model has N states, the right-hand side of the inequality above can be written as:

$$D_{i,j} = \sum_{n=1}^{N} |P_{in} - P_{jn}| \tag{7}$$

which is actually the *Manhattan distance* in the $\mathbb{R}^N$ space defined by the transition probabilities. To put it differently, we consider the states as N-valued vectors, where each one of the values is a transition probability to another state.

This shows that $D_{i,j} \geq E_{i,j}$. It is relatively straightforward to apply a clustering algorithm in order to identify K clusters such that the Manhattan distance $D_{i,j}$ is minimized for instances that belong to the same cluster. The minimization of $D_{i,j}$ will result in small values for $E_{i,j}$, and hence for the ϵ quantity in (4) as well.

4.2 The Clustering Algorithm

In order to obtain a partitioning of the state-space that minimizes the Manhattan distance for states in the same cluster, we have to apply a clustering algorithm. Such algorithms group the input data into *clusters* which minimize a distance metric between data in the same group. Typical clustering techniques, such as *K-means* or *Expectation-Maximization*, start from a randomly-picked initial solution and they perform a number of iterations until they converge to some optimum. Typically, multiple runs are required, as the solution obtained at each run is dependent on the initial randomly-picked solution.

In contrast, *spectral clustering* [19][21] implies that a dataset is partitioned depending on the eigenvectors of the *Laplacian* matrix, rather than on the local proximities of data-points. Concisely, the K eigenvectors that correspond to the largest K eigenvalues of the Laplacian are selected. The data is mapped to the rows of the $N \times K$ matrix formed by stacking these eigenvectors as columns. The clusters of data are well separated in this $\mathbb{R}^K$ space, meaning that it should be easy to identify a globally optimal clustering, in contrast to "conventional" clustering techniques whose solutions are only locally optimal. The algorithm of our choice is the one proposed by Ng et al in [21].

4.3 Quasi-Lumping

Assuming that we have a nearly optimal partition of the state-space, the next step is to construct a Markov chain that approximates the original model. Given some $N \times N$ lumpable matrix P with K equivalence classes $A_1, \ldots, A_K$, we define the corresponding $K \times K$ *lumped* matrix P' with entries:

$$P'_{ij} = \sum_{l \in A_j} P_{il} \tag{8}$$

where $i, j = 1, \ldots K$. We define a model to be *quasi-lumped* with respect to some matrix P, if it is lumped with respect to some matrix P^-, and $P = P^- + P^\epsilon$.

According to the definition of lumpability, the sums P'_{ij} in (8) for different states in the same class A_i will be the same. However, in the case of quasi-lumpable models they will only be approximately the same. The mean value is a reasonable approximator for populations characterized by almost the same

value, so we construct the quasi-lumped matrix $\hat{P}$ with entries:

$$\hat{P}_{ij} = \frac{\sum_{k \in A_i} \sum_{l \in A_j} P_{kl}}{|A_i|} \tag{9}$$

where $|A_i|$ denotes the number of states included in class A_i. It is evident that in the lumpable case, Equation (9) degrades to (8).

5 Compositional Aggregation

So far, we have discussed two ways to approximately aggregate a Markov chain. However, neither of these is directly applicable in practice, as they both require an explicit representation of the generator matrix. Instead, we attempt to reduce only parts of the model that are going to be combined in a compositional way.

For that reason, we can use a high-level modelling formalism such as PEPA [13], that enables us to model the system as a collection of cooperating components. The idea is to utilize a compositional representation of the underlying Markov chain of a PEPA model, or more accurately, a compositional representation of the corresponding generator matrix. This is actually possible by using the Kronecker form of a PEPA model, where the "global" generator matrix is defined in terms of the "partial" generator matrices of cooperating components combined via Kronecker algebra. It should be feasible to produce reduced versions of such partial generator matrices, and then combine them to obtain an approximately aggregated state-space.

As shown in [15], the generator matrix Q that corresponds to a PEPA model can be represented as a Kronecker product of terms in the following way:

$$Q = \bigoplus_{i=1}^{N} R_i + \sum_{a \in \mathcal{A}} r_a \times \left(\bigotimes_{i=1}^{N} P_{i,a} - \bigotimes_{i=1}^{N} \bar{P}_{i,a} \right) \tag{10}$$

where

- N is the number of components in the PEPA model.
- $\mathcal{A}$ is the set of shared actions.
- R_i is the rate matrix of i-th component based on its individual actions.
- r_a is the minimum *functional rate* of the shared action a over all components. The term 'functional rate' implies that the rate of an action depends on the state of one or more components. Equivalently, there is a single rate function $r_\alpha(C)$ that describes the apparent rate of action α for each state of component C. The minimum of the functional rates over all components C_i, $i = 1 \ldots N$ is defined as follows:

$$r_\alpha = \min(r_\alpha(C_1), r_\alpha(C_2), \ldots r_\alpha(C_N)) \tag{11}$$

- $P_{i,a}$ is the probability matrix of the i-th component for the shared action a. $\bar{P}_{i,a}$ is a diagonal matrix that ensures that the row sums of the corresponding probability matrix are zero, i.e. it is a valid generator matrix.

A useful observation regarding (10) is that any component C_i is described by two transition rate matrices: R_i which depends on its individual actions only, and $R_i^{(coop)} = \sum_{a \in \mathcal{A}} r_a P_{i,a}$ which cannot be determined, since we do not know the apparent rates of the cooperating components. If the set of shared actions is relatively small, we can expect that R_i will be much more dense than $R_i^{(coop)}$. If this condition holds, it should be reasonable to apply an approximate aggregation algorithm to R_i, in order to obtain a nearly optimal partition of this partial state-space.

This approach could be problematic though, as eliminating a shared action in a particular component may introduce deadlocks in its behaviour. For example, consider a component C_i with rate matrices:

$$R_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 \\ 3 & 0 & 6 & 0 & 0 \\ 0 & 3 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \end{bmatrix} \quad R_i^{(coop)} = \begin{bmatrix} 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 3 & 0 & 6 & 0 \\ 0 & 0 & 2 & 0 & 0 \end{bmatrix}$$

In the example above, R_i contains a deadlock at the fourth state, meaning that there is no non-trivial steady-state distribution over R_i in isolation, hence no way to compute the reversible process needed to apply the NCD-based approach, as described in Sect. 3.2. To solve this problem, we use the $\hat{R}_i$ matrix instead, which is constructed as in the following example:

$$\hat{R}_i = \begin{bmatrix} 0 & 0 & \varepsilon & 0 & 2 \\ 3 & 0 & 6 & 0 & 0 \\ 0 & 3 & 0 & \varepsilon & 4 \\ 0 & \varepsilon & 0 & \varepsilon & 0 \\ 0 & 0 & \varepsilon & 5 & 0 \end{bmatrix}$$

where $\varepsilon > 0$ is a small rate added to some transition for each shared action. Hence, if the original PEPA model contains no deadlocks, we can be sure that $\hat{R}_i$ will have no deadlocks either. By doing so, we obtain a partition of the component's state-space by using only a part of its behaviour. The ε rates added are equally distributed and therefore imply ignorance about the shared action rates.

The partitioning obtained using $\hat{R}_i$ is applied to both R_i and $R_i^{(coop)}$. Thus, the $N_i \times N_i$ partial generator matrix $Q_i = R_i + R_i^{(coop)}$ is approximated by the $K_i \times K_i$ matrix $Q_i' = R_i' + R_i'^{(coop)}$, where K_i is the number of partitions for the component C_i. Combining the reduced partial generator matrix Q_i using the Kronecker operations defined in (10), will result in a reduced global generator as well.

The state-space of a single sequential component does not usually involve more than a few states in typical models. It would be more effective if we could approximate components with a few hundreds of states instead. For that purpose, we apply clustering to cooperations of components rather than applying it to single sequential components. The cooperation rate matrix $R^{(coop)}$ of a non-sequential C component involves only actions that are shared with components

outside the cooperation. Hence, actions shared between sequential components included in the cooperation will only affect the individual rate matrix of C. In the context of this work, we apply the approximate reduction algorithms to populations of identical components.

6 A Multi-Scale Example

We compare the two different approaches for approximate Markov chain aggregation. The quasi-lumpability based approach described in Sect. 4 involves applying a clustering algorithm on the row entries of the transition probability matrix of a Markov chain. The NCD based approach discussed in Sect. 3 partitions the Markov chain according to the eigenvectors that correspond to the top eigenvalues of the probability matrix. Irreversible chains are handled by constructing a reversible process according to (3). For each one of the examples that follow, we explicitly note which components have been approximated and what compression ratio has been used. Once a nearly optimal partition of the state-space is obtained using either of the two methods, a reduced Markov chain is constructed as described in Sect. 4.3.

Eventually, we compare the transient and the steady-state behaviour of the initial model with those of the approximately aggregated models. The PRISM model checker [18], its sparse engine in particular, has been used for that purpose. The Jacobi algorithm has been applied for computing the steady-state distribution, and the uniformisation method for the transient probabilities. The experiments have been performed in an Intel® Quad-Core Xeon$^{\text{TM}}$ @ 3.20GHz PC running Linux.

At this point, we define a simple example to demonstrate the potential of the compositional approximate aggregation. We shall consider models featuring high-population components, as even simple model descriptions can lead to very large state-spaces. In particular, multi-scale models are of interest since more efficient approaches such as fluid flow approximation [14] are not as readily applicable, because they make an assumption of continuity which is strained at low population numbers. So we consider a peer-to-peer system that involves large numbers of peers that communicate with each other with the help of an indexing server, as described in the following PEPA model:

$$
\begin{aligned}
PeerA &\stackrel{def}{=} (localAction_A, r_{localA}).PeerA_{local} \\
&\quad + (lookup_B, \top).PeerA_{lookup} \\
PeerA_{local} &\stackrel{def}{=} (finish_A, r_{finishA}).PeerA \\
PeerA_{lookup} &\stackrel{def}{=} (cache_A, r_{cacheA}).PeerA_{local} \\
&\quad + (exchange, r_{exchangeA}).PeerA
\end{aligned}
$$

$$
\begin{aligned}
PeerB &\stackrel{def}{=} (localAction_B, r_{localB}).PeerB_{local} \\
&\quad + (lookup_A, \top).PeerB_{lookup} \\
PeerB_{local} &\stackrel{def}{=} (finish_B, r_{finishB}).PeerB \\
PeerB_{lookup} &\stackrel{def}{=} (cache_B, r_{cacheB}).PeerB_{local} \\
&\quad + (exchange, r_{exchangeB}).PeerB
\end{aligned}
$$

Our system involves two classes of peers which exchange data pairwise. Both types of peers have some local functionality and a shared activity called *exchange*. Moreover, a peer will have to look up other peers in an indexing server before proceeding to any data exchange.

$$\begin{aligned} Index &\stackrel{def}{=} (lookup_A, r_{lookupA}).Index_{busyA} \\ &\quad + (lookup_B, r_{lookupB}).Index_{busyB} \\ &\quad + (fail, r_{fail}).Index_{broken} \\ Index_{busyA} &\stackrel{def}{=} (refresh, r_{refresh}).Index \\ &\quad + (fail, r_{fail}).Index_{broken} \\ Index_{busyB} &\stackrel{def}{=} (refresh, r_{refresh}).Index \\ &\quad + (fail, r_{fail}).Index_{broken} \\ Index_{broken} &\stackrel{def}{=} (repair, r_{repair}).Index \end{aligned}$$

Table 1. Rate values used in the examples

Name	Value	Name	Value	Name	Value
r_{localA}	5	r_{localB}	2	$r_{lookupA}$	10
$r_{finishA}$	4	$r_{finishB}$	3	$r_{lookupB}$	10
r_{cacheA}	1	r_{cacheB}	2	r_{fail}	0.02
$r_{exchangeA}$	1	$r_{exchangeB}$	0.5	$r_{refresh}$	10
				r_{repair}	0.5

6.1 Compositional vs Global Aggregation

In this experiment we define a system small enough to compare the compositional approximate aggregation with a globally applied approach. The first system's structure is summarized in the following system equation, with cooperation sets $\mathcal{L} = \{exchange\}$ and $\mathcal{K} = \{lookup_A, lookup_B\}$.

$$System_{5:5:1} \stackrel{def}{=} PeerA[5] \underset{\mathcal{L}}{\bowtie} PeerB[5] \underset{\mathcal{K}}{\bowtie} Index$$

If we apply exact aggregation as described in [11], the number of states for the $PeerA[5]$ and $PeerB[5]$ components will be 21 (these would be 243 for each with no aggregation). Therefore, we distinguish the following cases:

i. $PeerA[5]$ and $PeerB[5]$ components are further reduced independently. The compression ratio used is 0.5 for both, resulting in a reduced chain of 400 states.
ii. Approximate aggregation is applied on the entire system's generator matrix. The compression ratio used was such that it results in a reduced chain of 400 states again.

Table 2. Execution Times for $System_{5:5:1}$

	Original	Quasi-Lumpability (Compositional)	NCD (Compositional)	Quasi-Lumpability (Global)	NCD (Global)
Approximation	-	0.15 sec	0.2 sec	205 sec	130 sec
PRISM Loading	2 sec	0.5 sec	0.5 sec	0.5 sec	0.5 sec
Transient Solution[a]	2.1 sec	0.6 sec	0.6 sec	0.6 sec	0.6 sec
Steady-State solution	0.2 sec	0.05 sec	0.05 sec	0.05 sec	0.05 sec
Total Time	4.3 sec	1.3 sec	1.35 sec	206.15 sec	131.15 sec
Number of states	1764	400	400	400	400

[a] 100 points: $0 \leq t \leq 2$

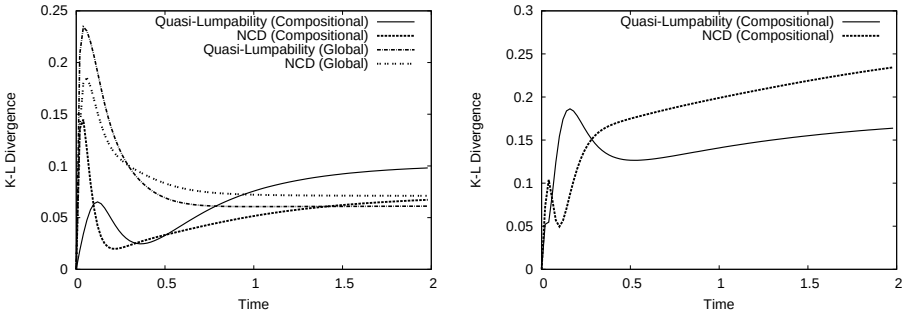
The *K-L divergence* is a very popular measure for comparing probability distributions. For two probability vectors $\boldsymbol{p}$ and $\boldsymbol{q}$, it is defined as:

$$KL(\boldsymbol{p}||\boldsymbol{q}) = \sum_i p_i \log \frac{p_i}{q_i} \tag{12}$$

Given a partition of the state-space with K classes, we define $\boldsymbol{p}$ as a K-valued vector containing the aggregated probabilities of the original system according to the partition of the state-space used. Then, $\boldsymbol{q}$ will be a K-valued vector containing the probabilities of the corresponding reduced model, which is produced by either the quasi-lumpability or the NCD approach. We want to see which one of the approximation approaches results in the lowest K-L divergence from the original state distribution.

The quasi-lumpability and the NCD based approaches have been applied in both a compositional and a global setting. Figure 1(a) summarizes the K-L divergences at different times t, for the four approximate aggregation methods. Judging by the K-L divergences, global aggregation does not appear to be far superior to the compositional approaches. Although there is no proof that this statement generalizes to every possible model, it seems reasonable to use compositional aggregation in order to produce a reasonable approximation of the original stochastic process. This argument is supported by Table 2, which summarizes the running times for aggregating and solving the model. As expected, compositional aggregation requires a very small initial cost to reduce the model, in contrast to the global case.

A second observation with respect to Fig. 1(a) is that neither the quasi-lumpability nor the NCD based approach seems to produce significantly more accurate results. In fact, the graphs are rather contradictory, as the global setting seems to favour quasi-lumpability, while in the compositional case NCD is the method that performs better. Figure 1(b) depicts the K-L divergences for $System_{10:20:2}$ of the next section. For this larger model, the compositionally applied quasi-lumpability approach is more accurate. Therefore, it seems reasonable to conclude that approximation accuracy is dependent on the properties of the model.



(a) $System_{5:5:1}$ (b) $System_{10:20:2}$

Fig. 1. Evolution of K-L divergences of various methods from the original state distribution

6.2 Approximation of Component Behaviour

This second example provides a more detailed view of component behaviour. The following system equation is considered, with cooperation sets $\mathcal{L} = \{exchange\}$ and $\mathcal{K} = \{lookup_A, lookup_B\}$.

$$System_{10:20:2} \stackrel{def}{=} PeerA[10] \underset{\mathcal{L}}{\bowtie} PeerB[20] \underset{\mathcal{K}}{\bowtie} Index[2]$$

If we apply exact aggregation as described in [11], the number of states for the $PeerA[10]$ component will be 66, while $PeerB[20]$ will have 231 states (these would be $59,049$ and $3,486,784,401$ states with no aggregation). Although neither of the components is particularly large, their combination results in a large state-space. However, it is relatively easy to further reduce $PeerA[10]$ and $PeerB[20]$ independently. The compression ratio used is 0.5 for both components.

This approximation of individual components results in significant reduction of the total state-space. As can be seen in Table 3, this reduction required only a small initial cost, while it resulted in a considerable decrease of the analysis time. A global reduction of the state-space would be practically infeasible for a models of such size. Figure 2(a) depicts the evolution of the average populations of the model components that have been reduced. Those figures seem to be reasonable approximations of the original model's average behaviour.

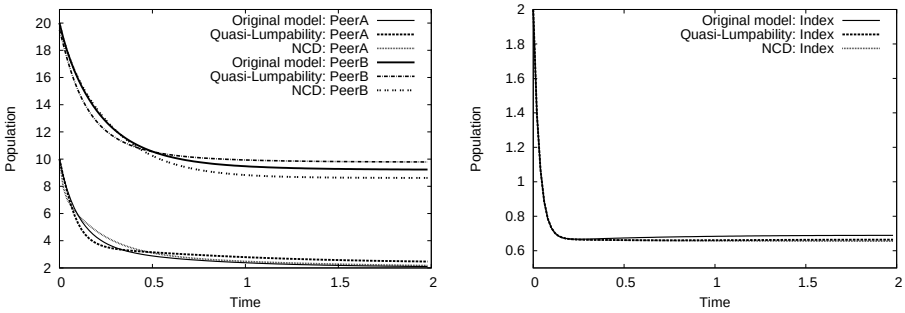
It would also be interesting though to look at the behaviour of the components that have not been approximated. Figure 2(b) depicts the evolution of the average *Index* populations. Both quasi-lumpability and NCD-based approach result in approximations very close to the original solution. This provides evidence that supports the claim that the behaviour of the unreduced components will be mostly unaffected, given a good partition of the state-space. Intuitively, we can approximately aggregate components whose behaviour is of minor importance and still obtain a very good approximation for the components that have not been approximated, which might be critical. In our example, if we were inter-

Table 3. Execution Times for $System_{10:20:2}$

	Original	Quasi-Lumpability	NCD
Approximation	-	1.2 sec	1.5 sec
PRISM Loading	433 sec	105 sec	105 sec
Transient Solution[b]	310 sec	93 sec	93 sec
Steady-State solution	21 sec	6 sec	6 sec
Total Time	764 sec	205.2 sec	205.5 sec
Number of states	152460	37950	37950

[b] 100 points: $0 \leq t \leq 2$

ested in the indexing servers' behaviour only, the approximation error would be negligible.



(a) *PeerA* and *PeerB* (b) *Index*

Fig. 2. Evolution of average populations for $System_{10:20:2}$

7 Conclusions

Although approximate Markov chain aggregation is not a new concept, it has not been particularly popular in the field of Markovian modelling, since an explicit representation of the transition matrix is typically required. In this paper, we have examined two different methods to approximately aggregate a Markov chain, and we have explored the potential of applying aggregation in a compositional way.

The traditional method for selecting a nearly optimal partition of the state-space makes use of the eigenstructure of the probability matrix. We have described this family of approaches as the NCD approach, since the eigenvectors convey information about parts of the state-space that are nearly completely decomposable. We have tried to define an alternative strategy of state-space

aggregation that relies on the concept of quasi-lumpability instead. More specifically, quasi-lumpability has been associated with the minimization of the $E_{i,j}$ measure between states in the same class. It has been shown that a simple clustering algorithm can be used to obtain an upper bound for this measure.

Intuitively, the quasi-lumpability approach should be superior, since a nearly completely decomposable system is essentially quasi-lumpable, but not vice-versa. Experimental results do not support this hypothesis though. In fact, it appears that some models favour the quasi-lumpability approach, while others the NCD approach. This can be attributed to the fact that the quasi-lumpability method is suboptimal, since it minimizes only an upper bound for $E_{i,j}$. A better approximation of the total $E_{i,j}$ error will be the subject of future work.

By using the Kronecker representation of PEPA models, we were able to reduce the local state-space of the labelled CTMCs that correspond to PEPA components. This practice resulted in a great reduction of the state-space size with a small initial cost for aggregating the PEPA components, in contrast with aggregating the entire Markov chain. The multi-scale example presented demonstrates the potential of compositional approximate aggregation in two ways. Firstly, the compositional approach resulted in a reasonable approximation of the original model, especially when compared to a global approach. Secondly, the error in the approximation of the unreduced components was found to be negligible, which means that critical components can be excluded from aggregation.

A final note on the applicability of our approach is that the approximated components are required to have a set of shared actions that is relatively small when compared to their set of individual actions. That would mean that the individual rate matrix is dense enough to apply a partitioning algorithm on it. Therefore, our approach is mostly applicable to models that can be decomposed to weakly dependent components. This is apparently related to the notion of quasi-separability, which has been applied to PEPA before [25]. A characterisation of the applicability of compositional aggregation in terms of quasi-separability is an interesting direction for future work.

Acknowledgments. The authors are supported by SynthSys, a Centre for Integrative Systems Biology (CISB) funded by BBSRC and EPSRC, reference BB/D019621/1.

References

1. P. Buchholz. Exact and ordinary lumpability in finite Markov chains. *Journal of Applied Probability*, 31(1):59–75, 1994.
2. A. Bušić and J. Fourneau. Bounds based on lumpable matrices for partially ordered state space. In *ICST Workshop on Tools for Solving Markov Chains*. ACM, 2006.
3. P. Courtois. Decomposability, instabilities, and saturation in multiprogramming systems. *Communications of the ACM*, 18(7):371–377, 1975.
4. D. Daly, P. Buchholz, and W. H. Sanders. Bound-preserving composition for Markov reward models. In *Quantitative Evaluation of Systems*, pages 243–252. IEEE Computer Society, 2006.

5. T. Dayar and W. Stewart. Quasi lumpability, lower-bounding coupling matrices, and nearly completely decomposable Markov chains. *SIAM Journal on Matrix Analysis and Applications*, 18(2):482–498, 1997.
6. K. Deng, Y. Sun, P. Mehta, and S. Meyn. An information-theoretic framework to aggregate a Markov chain. In *American Control Conference*, pages 731–736. IEEE Press, 2009.
7. P. Deuflhard, W. Huisinga, A. Fischer, and C. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Linear Algebra and its Applications*, 315(1-3):39–59, Aug. 2000.
8. J. Fourneau, M. Lecoz, and F. Quessette. Algorithms for an irreducible and lumpable strong stochastic bound. *Linear Algebra and its Applications*, 386:167–185, July 2004.
9. G. Franceschinis and R. Muntz. Bounds for quasi-lumpable Markov chains. *Performance Evaluation*, 20(1-3):223–243, May 1994.
10. G. Franceschinis and R. Muntz. Computing bounds for the performance indices of quasi-lumpable stochastic well-formed nets. *IEEE Transactions on Software Engineering*, 20(7):516–525, July 1994.
11. S. Gilmore, J. Hillston, and M. Ribaudo. An efficient algorithm for aggregating PEPA models. *IEEE Transactions on Software Engineering*, 27(5):449–464, May 2001.
12. D. Hartfiel. On the structure of stochastic matrices with a subdominant eigenvalue near 1. *Linear Algebra and its Applications*, 272(1-3):193–203, Mar. 1998.
13. J. Hillston. *A compositional approach to performance modelling*. Cambridge University Press, 1996.
14. J. Hillston. Fluid flow approximation of PEPA models. In *Quantitative Evaluation of Systems*, pages 33–42. IEEE Computer Society, 2005.
15. J. Hillston and L. Kloul. An efficient Kronecker representation for PEPA models. In *Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification*, pages 120–135. Springer-Verlag, 2001.
16. A. Jensen. Markoff chains as an aid in the study of Markoff processes. *Skandinavisk Aktuarietidskrift*, 36:87–91, 1953.
17. J. Kemeny and J. Snell. *Finite Markov Chains*. Springer, 1976.
18. M. Kwiatkowska, G. Norman, and D. Parker. PRISM: probabilistic model checking for performance and reliability analysis. *ACM SIGMETRICS Performance Evaluation Review*, 36(4):40–45, 2009.
19. J. Malik and J. Shi. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
20. C. D. Meyer. Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Review*, 31(2):240–272, 1989.
21. A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14(1):849–856, 2001.
22. P. Pokarowski. Uncoupling measures and eigenvalues of stochastic matrices. *Journal of Applied Analysis*, 4(2):259–267, Dec. 1998.
23. T. Runolfsson and Y. Ma. Model reduction of nonreversible Markov chains. In *IEEE Conference on Decision and Control*, pages 3739–3744. IEEE, 2008.
24. M. Smith. Compositional abstractions for long-run properties of stochastic systems. In *Quantitative Evaluation of Systems*, pages 223–232. IEEE Computer Society, 2011.
25. J. Thomas, Nigel and Bradley. Analysis of non-product form parallel queues using Markovian process algebra. In *Network performance engineering*, pages 331–342. Springer-Verlag, 2011.