

# Iteration at Different Levels: Multi-Level Methods for Structured Markov Chains

Peter Buchholz  
Informatik IV, Universität Dortmund  
D-44221 Dortmund, Germany  
email: peter.buchholz@udo.edu

## 1 Introduction

Markov models are an established model type to describe systems from various application areas, including computer systems and networks, communication systems, biological systems, social systems and are also used to compute the page rank of Web pages and related measures. We consider here Markov chains with a finite state space  $\mathcal{RS} = \{0, \dots, n-1\}$ . Very often the stationary distribution is needed resulting in the solution of the set of equations

$$\boldsymbol{\pi}\mathbf{Q} = \mathbf{0} \text{ and } \boldsymbol{\pi}\mathbf{e}^T = 1 \quad (1)$$

in the continuous case and in the eigenvector computation

$$\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi} \quad (2)$$

in the discrete time case [23]. In the former case  $\mathbf{Q}$  is assumed to be an irreducible generator matrix and in the latter case,  $\mathbf{P}$  is an irreducible stochastic matrix. It is well known that  $\mathbf{Q} = (\mathbf{P} - \mathbf{I})$  transforms a discrete time model into a continuous time model with the same stationary distribution and  $\mathbf{P} = \mathbf{Q}/\alpha + \mathbf{I}$  with  $\alpha \geq \max_{x \in \mathcal{RS}} |\mathbf{Q}(x, x)|$  transforms a continuous time model into a discrete time model with the same stationary distribution. We will consider in this short note Markov processes in continuous time and since we only consider stationary analysis, the transformation from discrete to continuous time is straightforward.

The major problem of computation of the stationary distribution for real examples is the size of the state space which can be several hundreds of millions or even billions of states. Thus, space and time efficient solution techniques are of major importance. The first step in developing more efficient analysis techniques is usually the finding of some structure for the state space. A common approach which naturally fits to many descriptions of real world examples is a multi-dimensional structure. Let  $\mathcal{J} = \{1, \dots, J\}$  be the set of component indices and let  $\mathcal{RS}^{(j)} = \{0, \dots, n^{(j)}-1\}$  be the set of states

in dimension  $j$  or the states of component  $j$ , then  $\mathcal{PS} = \times_{j=1}^J \mathcal{RS}^{(j)}$  is the set of potentially reachable states in the composed model and if the transition rates are composable and  $\mathcal{PS}$  equals  $\mathcal{RS}$  the set of reachable states, then the generator matrix can be represented as

$$\hat{\mathbf{Q}} = \sum_{t \in \mathcal{T}} \lambda_t \left( \bigotimes_{j=1}^J \mathbf{E}_t^{(j)} - \bigotimes_{j=1}^J \mathbf{D}_t^{(j)} \right) \quad (3)$$

where  $\mathcal{T}$  is the set of model transitions and  $\lambda_t$  is the transition rate of transition  $t$  which might as well be included in one of the matrices  $\mathbf{E}_t^{(j)}$ .  $\mathbf{E}_t^{(j)}$  is the matrix describing the effect of transition  $t$  at component/dimension  $j$  and  $\mathbf{D}_t^{(j)} = \text{diag}(\mathbf{E}_t^{(j)} \mathbf{e}^T)$  is a diagonal matrix including the row sums of  $\mathbf{E}_t^{(j)}$ . Matrix  $\hat{\mathbf{Q}}$  can in principle be used for the computation of the transient or stationary solution vector as first noticed for a slightly more general class of automata networks by Plateau [21]. In particular, it is possible to compute the product of a vector and the descriptor  $\hat{\mathbf{Q}}$  using the Kronecker representation from (3) rather than building the huge matrix.

The disadvantage of (3) is that  $\mathcal{PS}$  is usually much larger than  $\mathcal{RS}$  since it contains unreachable states. To avoid this problem an additional dimension is introduced resulting in hierarchical matrices [6, 11].

For a hierarchical representation the state space  $\mathcal{RS}$  is decomposed into global macro states such that matrix  $\mathbf{Q}$  is structured into blocks

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_{1,1} & \cdots & \mathbf{Q}_{1,\tilde{n}} \\ \vdots & \ddots & \vdots \\ \mathbf{Q}_{\tilde{n},1} & \cdots & \mathbf{Q}_{\tilde{n},\tilde{n}} \end{pmatrix} \quad (4)$$

where  $\tilde{n}$  is the number of global macro states. Diagonal submatrices describe transitions inside global macro states and non diagonal matrices transitions between global macro states. According to the global macro states, each state space  $\mathcal{RS}^{(j)}$  is decomposed into subsets of states that belong to a global macro state. We denote these subsets as local macro states. A submatrix  $\mathbf{Q}_{k,l}$  can then be represented as

$$\mathbf{Q}_{k,l} = \left( \sum_{t \in \mathcal{T}} \lambda_t \left( \bigotimes_{j=1}^J \mathbf{E}_t^{(j)}[j_k, j_l] - \delta(k=l) \bigotimes_{j=1}^J \mathbf{D}_t^{(j)}[j_k, j_l] \right) \right). \quad (5)$$

Here  $j_k$  is the local macro state of component  $j$  in the global macro state  $k$  and submatrices include transitions between states belonging to the local macro states  $j_k$  and  $j_l$ . The representation is very compact since involved matrices are of order  $n^{(j)}(j_k)$  rather than  $\prod_{j=1}^J n^{(j)}(j_k)$  where  $n^{(j)}(j_k)$  is the number of detailed states in the local macro state  $j_k$  belonging to global macro state  $k$ . Hierarchical matrices can be generated automatically at the model [10, 12] or state space level [8]. The hierarchical matrix representation

can be extended to discrete time models as shown in [7] for queuing networks in discrete time.

It is straightforward to realize iterative solution techniques in conjunction with the compact matrix representation [1]. However, due to the state space size the solution time is often prohibitive such that more efficient approaches are required.

## 2 Multi Level Techniques

To speed up convergence of iterative methods so called multi level techniques have been proposed in various application areas [13]. The idea of all these approaches is to define a set of related iteration schemes considering different levels of detail, perform iterations at each level and transfer the improved or smoothed iteration vector from one level to another. For Markov chain problems, the approaches are often based on aggregation-disaggregation and use only two level [16, 20, 23]. However, also some aggregation/disaggregation approaches for more than two levels exist [2, 9, 15, 24]. In the sequel we present an extended version of the approaches proposed in [2, 9].

For any multi-level method, the following decisions have to be made:

1. How to define levels and how to transfer results from one level to another.
2. How to perform iterations at the different levels and how to define a complete iteration scheme.

We begin with the definition of different levels and their interaction and present afterwards the complete algorithm. Finally very briefly some results are shown.

### 2.1 Definition of Levels

For models resulting from the composition of components every state from  $\mathcal{RS}$  can be represented by a  $J$ -dimensional vector  $\mathbf{x} = (\mathbf{x}(0), \dots, \mathbf{x}(J))$  such that  $\mathbf{x}(j) \in \mathcal{RS}^{(j)}$ . We define the most detailed or finest level by distinguishing all states from  $\mathcal{RS}$ . A natural way of defining coarser levels is to use subsets  $\mathcal{C} \subset \mathcal{J}$  such that two states  $\mathbf{x} = (\mathbf{x}(0), \dots, \mathbf{x}(J))$  and  $\mathbf{y} = (\mathbf{y}(0), \dots, \mathbf{y}(J))$  are indistinguishable at the coarser level if and only if  $\mathbf{x}(i) = \mathbf{y}(i)$  for all  $i \in \mathcal{C}$  and  $\mathbf{x}(0) = \mathbf{y}(0)$ . Observe that the block level, component zero in each vector is not included in any subset such that only states from the same block are possibly mapped on the same aggregated state. Thus,  $\mathcal{C} = \emptyset$  results in a mapping where all states from one block are mapped on the same state.

To generalize the approach define  $\mathcal{C} \subset \mathcal{D} \subseteq \mathcal{J}$ . We use the subset as index to describe vectors and matrices at some level. I.e.,  $\mathcal{RS}_{\mathcal{C}}$ ,  $\mathbf{Q}_{\mathcal{C}}$  and  $\pi_{\mathcal{C}}$

are the state space, generator matrix and stationary solution vector of the Markov chain at the level defined by the detailed consideration of component states from the set  $\mathcal{C}$ . A mapping from  $\mathcal{RS}_{\mathcal{D}}$  into  $\mathcal{RS}_{\mathcal{C}}$  is defined by a matrix  $\mathbf{R}_{\mathcal{D},\mathcal{C}}$  with  $|\mathcal{RS}_{\mathcal{D}}|$  rows and  $|\mathcal{RS}_{\mathcal{C}}|$  columns. Such that for two states  $\mathbf{x}_{\mathcal{D}} = [\mathbf{x}(i)]_{i \in \mathcal{D}}$  and  $\mathbf{y}_{\mathcal{C}} = [\mathbf{y}(j)]_{j \in \mathcal{C}}$ ,  $\mathbf{R}_{\mathcal{D},\mathcal{C}}(\mathbf{x}_{\mathcal{D}}, \mathbf{y}_{\mathcal{C}}) = 1$  if  $\mathbf{x}(j) = \mathbf{y}(j)$  for all  $j \in \mathcal{C}$  and  $\mathbf{R}_{\mathcal{D},\mathcal{C}}(\mathbf{x}_{\mathcal{D}}, \mathbf{y}_{\mathcal{C}}) = 0$  otherwise. Observe that  $\mathbf{R}_{\mathcal{D},\mathcal{C}}$  has row sum 1 and contains elements 0 or 1. Furthermore  $\mathbf{R}_{\mathcal{J},\mathcal{C}} = \mathbf{R}_{\mathcal{J},\mathcal{D}}\mathbf{R}_{\mathcal{D},\mathcal{C}}$ . Matrix  $\mathbf{R}$  is denoted as restriction operator. It is independent of the current solution or iteration vector. Let  $r_{\mathcal{C},\mathcal{D}}^{-1}(\mathbf{y}) = \{\mathbf{x} | \mathbf{R}_{\mathcal{D},\mathcal{C}}(\mathbf{x}, \mathbf{y}) = 1\}$  for  $\mathbf{y} \in \mathcal{RS}_{\mathcal{C}}$ .

To map results from a coarse level  $\mathcal{RS}_{\mathcal{C}}$  to a finer level  $\mathcal{RS}_{\mathcal{D}}$ , a prolongation or projection operator is defined which is a matrix  $\mathbf{P}_{\mathcal{C},\mathcal{D}}$  with  $|\mathcal{RS}_{\mathcal{C}}|$  rows and  $|\mathcal{RS}_{\mathcal{D}}|$  columns.  $\mathbf{P}_{\mathcal{C},\mathcal{D}}$  is defined as a non negative matrix with row sum 1 such that  $\mathbf{P}_{\mathcal{C},\mathcal{D}}(\mathbf{y}, \mathbf{x}) > 0$  implies  $\mathbf{R}_{\mathcal{D},\mathcal{C}}(\mathbf{x}, \mathbf{y}) = 1$ . The relation  $\mathbf{R}_{\mathcal{D},\mathcal{C}}\mathbf{P}_{\mathcal{C},\mathcal{D}} = \mathbf{I}$  holds. There are two possibilities to define projection operators. The first is to define a constant matrix (see e.g. [24]), the second is to define matrix  $\mathbf{P}$  depending the current solution vector at  $\mathcal{RS}_{\mathcal{D}}$ . Let  $\mathbf{p}_{\mathcal{D}}$  some estimate of the solution vector at this level, then

$$\mathbf{P}_{\mathcal{C},\mathbf{p}_{\mathcal{D}}}(\mathbf{y}, \mathbf{x}) = \begin{cases} \mathbf{p}_{\mathcal{D}}(\mathbf{x}) / \sum_{\mathbf{z} \in r_{\mathcal{C},\mathcal{D}}^{-1}(\mathbf{y})} \mathbf{p}_{\mathcal{D}}(\mathbf{z}) & \text{if } \mathbf{x} \in r_{\mathcal{C},\mathcal{D}}^{-1}(\mathbf{y}), \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

With the restriction and prolongation operators Markov chains at the different levels can be defined. Assume that  $\mathbf{p}_{\mathcal{D}}$  and  $\mathbf{Q}_{\mathcal{D}}$  are an approximation of the stationary vector and the generator matrix of a Markov chain with state space  $\mathcal{RS}_{\mathcal{D}}$ . Then a mapping of the approximation and the generator matrix over state space  $\mathcal{RS}_{\mathcal{C}}$  can be defined as follows.

$$\mathbf{p}_{\mathcal{C}} = \mathbf{p}_{\mathcal{D}}\mathbf{R}_{\mathcal{D},\mathcal{C}} \quad \text{and} \quad \mathbf{Q}_{\mathcal{C}} = \mathbf{P}_{\mathcal{C},\mathcal{D}}\mathbf{Q}_{\mathcal{D}}\mathbf{R}_{\mathcal{D},\mathcal{C}} \quad (7)$$

Instead of  $\mathbf{P}_{\mathcal{C},\mathcal{D}}$ , also  $\mathbf{P}_{\mathcal{C},\mathbf{p}_{\mathcal{D}}}$  may be used. The following two theorems are proved in [5].

**Theorem 2.1** *If  $\mathbf{Q}_{\mathcal{D}}$  is an irreducible generator matrix, the prolongation matrix contains one non zero element in each column and  $\mathcal{C} \subset \mathcal{D}$ , then  $\mathbf{Q}_{\mathcal{C}}$  is an irreducible generator matrix.*

*If  $\mathbf{p}_{\mathcal{D}} > \mathbf{0}$ , then  $\mathbf{P}_{\mathcal{C},\mathbf{p}_{\mathcal{D}}}$  contains one non zero element per column.*

Thus, if  $\mathbf{Q}_{\mathcal{J}}$  is irreducible, we only need to assure  $\mathbf{p} > \mathbf{0}$  and use a prolongation matrix depending on  $\mathbf{p}$  as defined in (6) at each level to generate irreducible generator matrices at each level.

**Theorem 2.2** *If  $\pi_{\mathcal{D}}$  is the stationary solution vector of the Markov chain with generator matrix  $\mathbf{Q}_{\mathcal{D}}$  and  $\mathbf{Q}_{\mathcal{C}}$  is computed using prolongation matrix  $\mathbf{P}_{\mathcal{C},\pi_{\mathcal{D}}}$ , then  $\pi_{\mathcal{D}}\mathbf{R}_{\mathcal{D},\mathcal{C}}$  is the stationary solution vector of the Markov chain with generator matrix  $\mathbf{Q}_{\mathcal{C}}$ .*

If  $\mathbf{Q}_{\mathcal{J}}$  has a Kronecker representation as in (4) and (5), then every aggregated matrix  $\mathbf{Q}_{\mathcal{D}}$  has a Kronecker representation that includes only matrices for automata with indices from  $\mathcal{D}$  and some vectors. For details see [2, 9].

## 2.2 The Iteration Scheme

In general there are different possibilities to define multi-level iterations. Using the notation of [5] the iteration matrix of the multi-level approach can be represented in a recursive form. Let  $\mathcal{D}_1 \subset \mathcal{D}_2 \subset \dots \subset \mathcal{D}_K = \mathcal{J}$  be a set of nested component index sets. The following recursive description defines the multi-level iteration matrix for  $k > 1$ .

$$\mathbf{T}_{\mathcal{D}_k}^{ML} = (\mathbf{T}_{\mathcal{D}_k})^{\nu_1} \mathbf{R}_{\mathcal{D}_k, \mathcal{D}_{k-1}} \mathbf{T}_{\mathcal{D}_{k-1}}^{ML} \mathbf{P}_{\mathcal{D}_{k-1}, \mathbf{p}_{\mathcal{D}_k}} (\mathbf{T}_{\mathcal{D}_k})^{\nu_2} \quad (8)$$

Here  $\mathbf{T}_{\mathcal{D}_k}$  is the iteration matrix at level  $\mathcal{D}_k$  which might be the iteration matrix of the Power method (i.e.,  $\mathbf{T}_{\mathcal{D}_k} = \mathbf{Q}_{\mathcal{D}_k} / \alpha + \mathbf{I}$  for  $\alpha \geq \max_{x \in \mathcal{R}\mathcal{S}_{\mathcal{D}_k}} |\mathbf{Q}_{\mathcal{D}_k}(x, x)|$ ), JOR or SOR.  $\nu_1$  and  $\nu_2$  are the numbers of pre- and post-iterations and  $\mathbf{p}_{\mathcal{D}}$  is the iteration vector after the pre-iterations have been performed. For the coarsest level  $\mathcal{D}_1$  we define  $\mathbf{T}_{\mathcal{D}_1}^{ML}$  either as  $\mathbf{T}_{\mathcal{D}_1}^{\nu_0}$  or as  $\mathbf{e}^T \boldsymbol{\pi}_{\mathcal{D}_1}$  where  $\boldsymbol{\pi}_{\mathcal{D}_1}$  is the stationary vector of  $\mathbf{Q}_{\mathcal{D}_1}$ . An iteration step  $\mathbf{p}_j^{(k)} = \mathbf{p}_j^{(k-1)} \mathbf{T}_{\mathcal{J}}^{ML}$  is denoted as a cycle.

Equation (8) describes the V-cycle of the multi-level method using a prolongation operator that depends on the current solution iteration vector. It is, of course, also possible to use a constant prolongation matrix. Similarly, W- oder F-cycles can be defined for the multi level approach.

Apart from the choice of a cycle type and a prolongation operator, there are still two additional issues. First, how to define nested sets of component indices and second, how to choose  $\nu_1$  and  $\nu_2$ . A natural way of defining levels is to remove one component in each step. This results in  $J$  levels. In principle  $J!$  different orderings exist in which components can be removed and the ordering may even change from cycle to cycle. Typical orderings are fixed orderings where in every cycle indices are removed in the same order, a circular ordering where between the cycles component indices are cyclically shifted and dynamic ordering where indices are reordered between two cycles according to an estimate of the local error in the corresponding dimension. The number of pre- and post iterations can be fixed or it can depend on the convergence behavior. In the latter case let  $\mathbf{r}_{\mathcal{D}} = \mathbf{p}_{\mathcal{D}} \mathbf{Q}_{\mathcal{D}}$  the residual vector at level  $\mathcal{D}$ . Then  $\nu_1$  and  $\nu_2$  may be set dynamically such that in each cycle at each level the norm of the residual vector is reduced. Since  $\mathbf{Q}_{\mathcal{D}}$  is an irreducible generator matrix under the conditions given above, the Power method assures that convergence at every level can be reached.

### 2.3 Example Results

Since a proof of the convergence speed of the multi-level approach seems to be hard to achieve, one has to use empirical results to compare it with other solution techniques. A fairly large set of example results is presented in [4]. Here we only show results for two examples and summarize some general observations.

Method	Iterations	CPU time in sec.	$\ \mathbf{r}\ _\infty$
Power	79,060	448.3	$9.988e - 9$
ML(Power, fixed, 1, 1)	164	3.1	$9.019e - 9$
SOR	9590	92.1	$9.928e - 9$
ML (SOR, fixed, 1, 1)	20	0.5	$8.439e - 9$
BiCGStab (Pre. BSOR)	390	9.2	$1.946e - 8$

Table 1: Results for different solution methods applied to the example with *ncd* property.

The first example is a queuing network example with loosely coupled sub-models. A job performs a cycle of operations at different queues and changes the component at the end of a cycle with a probability 1%. This model has a mild NCD-property [23]. Our specific configuration consists of 4 components and results in a Markov chain with about 50,000 states. Table 1 contains the results for this example. We show the required number of iterations and count for the multi-level methods only outer iterations. Furthermore, the CPU time and the residual norm after termination of the algorithms is presented. Five different solution techniques are compared. The Power method and SOR as conventional iterative methods [23], BiCGStab with a BSOR preconditioner [3] as a projection method and two multi-level variants using the Power method and SOR as smoother and 1 pre- and post-iteration per cycle. From the results it is obvious that the multi-level methods clearly outperform the other solution methods.

Method	Iterations	CPU time in sec.	$\ \mathbf{r}\ _\infty$
Power	3,990	3502.0	$9.263e - 9$
ML(Power, circular, 1, 1)	892	1227.0	$9.734e - 9$
SOR	140	162.5	$3.062e - 9$
ML (SOR, circular, 1, 1)	50	87.8	$2.450e - 9$
BiCGStab (Pre. BSOR)	40	210.6	$5.886e - 9$

Table 2: Results for different solution methods applied to the courier protocol example.

The second example we consider is a stochastic Petri net model describing two layers of a simple client server protocol [25]. The configuration we analyzed results in a Markov chain with 1,632,600 states. For this example

the different components are strongly coupled. However, as one can see from the results in table 2, multi-level methods with a circular ordering of components to be aggregated still outperform other solution methods although the difference is not as large as for the previous example.

From our fairly comprehensive set of example runs in [4] we draw the following conclusions:

- Multi-level methods are for almost all models we analyzed the fastest solvers. The speedup of multi-level methods compared to other iterative methods can be several orders of magnitude if components are loosely coupled.
- A small and fixed number of pre- and post-iterations seems to be the best choice.
- SOR is often the best choice as a smoother.
- The difference between the three cycle types is relatively small.
- Among the ordering of aggregation we tried, a fixed ordering seems to be the most robust choice, whereas a circular or dynamic ordering depending on the local residual is often more efficient.
- A prolongation operator depending on the current iteration vector gave much better results than a static prolongation operator. With a static prolongation operator, the multi-level approach was often not faster than the basic iteration method without multi-level steps.
- Multi-level approaches may be used as preconditioners for prolongation methods. Our experience is that the approach does not work with prolongation operators depending on the iteration vector and for fixed projection operators we obtained no speed up compared to the projection method without preconditioning.

### 3 What about Page Rank?

The matrices that are used in the previous section result from a multi dimensional description such that levels follow naturally from the definition of the model. For most Markov chains for information retrieval including the Google matrix [17] such a structure does not exist. However, one can try to approximate the matrix by a structured matrix. Some results about reordering the Google matrix to a form with strongly coupled blocks exist [18] and also first applications of two level methods to the compute page rank exist [19].

To apply the multi level approach presented here to general stochastic matrices  $\mathbf{P}$ , one has to define a block decomposition such that every block

can be approximated by a sum of Kronecker products of matrices. I.e., the goal is to find a substochastic matrix  $\mathbf{A}$  such that  $\mathbf{A}$  has a Kronecker representation similar (4), (5) and  $\mathbf{P} = \mathbf{A} + \mathbf{D}$  where the elements of  $\mathbf{D}$  are non negative and small compared to the elements of  $\mathbf{A}$ . Usually  $\mathbf{A}$  needs not be irreducible since  $\mathbf{P}$  is not irreducible. For the computation of the stationary vector, the matrix  $\alpha\mathbf{P} + (1 - \alpha)\mathbf{e}\mathbf{v}^T$  is used where  $\mathbf{v}$  is the personalization vector (see [17] for details).

If matrix  $\mathbf{A}$  has been found, then the multi-level approach can be applied using the following iteration scheme.

$$\mathbf{p}^{(k)}(\mathbf{A} - \mathbf{I})^{ML} = \mathbf{b} \text{ where } \mathbf{b} = -\mathbf{p}^{(k-1)}\mathbf{D} \quad (9)$$

In contrast to the solution of structured Markov chains the multi level approach now has to be applied for a system with non zero right hand side which can be done without problems.

It remains to find an approximate matrix  $\mathbf{A}$  which is the hardest problem. The goal is to find for some submatrix  $\mathbf{B}$  of  $\mathbf{A}$  two matrices  $\mathbf{B}_1 \otimes \mathbf{B}_2$  such that  $\mathbf{B} \approx \mathbf{B}_1 \otimes \mathbf{B}_2$ . Several approaches exist to approximate matrices by Kronecker products of matrices [14, 22] and may be applied for this purpose.

## 4 Conclusions and Open Research Questions

Multi-level approaches as an adaption of algebraic multi-grid methods to continuous time Markov chains have shown to be very efficient solvers which outperform other iterative techniques for most examples. However, multi-level methods define a class of methods rather than a single method. A large number of parameters is available for these methods which all have some influence on the performance of the resulting solver. Currently, it is still an open research question how to set these parameters to obtain an optimal or nearly optimal solver for a given problem. To exploit the full potential of multi-level methods, adaptive steps have to be integrated to set the parameters of the method according to the observed convergence speed.

It has been shown that a multi level structure is naturally given for most models for performance or dependability analysis, but it is usually not available for Markov chains resulting from Web graphs. To exploit structured solution methods also to those Markov chains, the resulting matrices have to be represented or approximated by a sum of Kronecker products of small matrices. It has been outlined how such a representation may be found, but algorithms for this purpose are still missing.

## References

- [1] P. Buchholz. Structured analysis approaches for large Markov chains. *Applied Numerical Mathematics*, 31(4):375–404, 1999.

- [2] P. Buchholz. Multi-level solutions for structured Markov chains. *SIAM Journal on Matrix Methods and Applications*, 22(2):342–357, 2000.
- [3] P. Buchholz and T. Dayar. Block SOR preconditioned projection methods for Kronecker structured Markovian representations. *SIAM Journal on Scientific Computing*, 26:1289–1313, 2005.
- [4] P. Buchholz and T. Dayar. Experimental comparison of structured analysis techniques for large Markov chains. Technical Report Forthcoming Technical Report, Universität Dortmund, Fachbereich Informatik, 2007.
- [5] P. Buchholz and T. Dayar. On the convergence of a class of multilevel methods for large sparse Markov chains. *SIAM Journal on Matrix Methods and Applications*, to appear 2007.
- [6] Peter Buchholz. A hierarchical view of gcsdns and its impact on qualitative and quantitative analysis. *J. Parallel Distrib. Comput.*, 15(3):207–224, 1992.
- [7] Peter Buchholz. On the exact and approximate analysis of hierarchical discrete time queueing networks. In Heinz Beilner and Falko Bause, editors, *MMB*, volume 977 of *Lecture Notes in Computer Science*, pages 150–164. Springer, 1995.
- [8] Peter Buchholz. Hierarchical structuring of superposed gspns. *IEEE Trans. Software Eng.*, 25(2):166–181, 1999.
- [9] Peter Buchholz and Tugrul Dayar. Comparison of multilevel methods for kronecker-based markovian representations. *Computing*, 73(4):349–371, 2004.
- [10] Peter Buchholz and Peter Kemper. On generating a hierarchy for gspn analysis. *SIGMETRICS Performance Evaluation Review*, 26(2):5–14, 1998.
- [11] Peter Buchholz and Peter Kemper. Kronecker based matrix representations for large markov models. In Christel Baier, Boudewijn R. Haverkort, Holger Hermanns, Joost-Pieter Katoen, and Markus Siegle, editors, *Validation of Stochastic Systems*, volume 2925 of *Lecture Notes in Computer Science*, pages 256–295. Springer, 2004.
- [12] Javier Campos, Susanna Donatelli, and Manuel Silva. Structured solution of asynchronously communicating stochastic modules. *IEEE Trans. Software Eng.*, 25(2):147–165, 1999.
- [13] Robert D. Falgout. An introduction to algebraic multigrid. *Computing in Science and Engineering*, 8(6):24–33, 2006.

- [14] M. P. Friedlander and K. Hatz. Computing nonnegative tensor factorizations. Technical Report TR-2006-21, Dep. of Comp. Sci., University of British Columbia, 2006.
- [15] Graham Horton and Scott T. Leutenegger. A multi-level solution algorithm for steady-state markov chains. In *SIGMETRICS*, pages 191–200, 1994.
- [16] U. Krieger. Numerical solution of large finite Markov chains by algebraic multigrid techniques. In W. J. Stewart, editor, *Computations with Markov chains*, pages 403–424. Kluwer Academic Publishers, 1995.
- [17] A. Langville and C. D. Meyer. A survey of eigenvector methods for web information retrieval. *SIAM Review*, 47(1):135–161, 2005.
- [18] A. N. Langville and C. D. Meyer. A reordering for the PageRank problem. Technical report, Dep. of Mathematics, N. Carolina State University, 2004.
- [19] C. Pan-Chi Lee, G. H. Golub, and S. A. Zenios. A fast two-stage algorithm for computing PageRank and its extensions. Technical Report SCCM-03-15, Dep. of Comp. Sci., Stanford University, 2003.
- [20] I. Marek and P. Mayer. Convergence analysis of an iterative aggregation/disaggregation method for computing the probability vector of stochastic matrices. *Numerical Linear Algebra with Applications*, 5(4):253–274, 1998.
- [21] Brigitte Plateau. On the stochastic structure of parallelism and synchronization models for distributed algorithms. In *SIGMETRICS*, pages 147–154, 1985.
- [22] A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In Luc De Raedt and Stefan Wrobel, editors, *ICML*, pages 792–799. ACM, 2005.
- [23] W. J. Stewart. *Introduction to the numerical solution of Markov chains*. Princeton University Press, 1994.
- [24] E. Virnik. An algebraic multigrid preconditioner for a class of singular M-matrices. Preprint 03-2006, TU Berlin, 2006.
- [25] C. M. Woodside and Y. Li. Performance Petri net analysis of communications protocol software by delay equivalent aggregation. In *Proc. 4th Int. Workshop on Petri Nets and Performance Models*, pages 64–73. IEEE CS Press, 1991.