Efficient Solutions for an Approximation Technique for the Transient Analysis of Markovian Models

Rosa Carmo, Edmundo de Souza E Silva, Raymond Marie

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Efficient Solutions for an Approximation Technique for the Transient Analysis of Markovian Models

Rosa M. L. R. Carmo, Edmundo de Souza e Silva

and Raymond Marie

N ° 3055

Novembre 1996

THÈME 1

ISSN 0249-6399

INRIA
RENNES

Efficient Solutions for an Approximation Technique for the Transient Analysis of Markovian Models

Rosa M. L. R. Carmo, * Edmundo de Souza e Silva*
and Raymond Marie †

Thème 1 — Réseaux et systèmes
Projet model

Rapport de recherche n ° 3055 — Novembre 1996 — 19 pages

Abstract: One of the most widely used technique to obtain transient measures is the uniformization method. However, although uniformization have many advantages, the computational cost required to calculate transient probabilities are very large for stiff models. We study efficient solutions that can be applied to an approximate method developed for calculating transient state probabilities of Markov models, and cumulative expected reward measures over a finite interval. The method our work is based on, approximates the state probabilities at time t by the state probabilities calculated at a random time with Erlangian distribution. The original method requires an inversion of a matrix obtained from the state transition rate matrix, which destroys special structures such as sparseness and banded matrices. This precludes the use of the technique for large models. In our work we propose efficient solutions that can take advantage of special structures. Finally, we present examples which show that the proposed technique is computationally very efficient for stiff models when compared with uniformization.

Key-words: Markovian Models, Transient Solution, uniformization, stiffness

(Résumé : tsvp)

This work was partially supported by the Inria-CNPq(Brazil) agreement.

* Federal University of Rio de Janeiro, Computer Science Department, NCE, COPPE/Systems, Cx.P. 2324, Rio de Janeiro, RJ 20001-970, Brazil
† {Raymond.Marie}@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 02 99 84 71 00 – Télécopie : (33) 02 99 84 71 71

Solutions efficaces relatives à une méthode approchée pour l'analyse transitoire de modèles markoviens

Résumé : L'une des techniques les plus utilisées pour obtenir des résultats sur le comportement transitoire des modèles markoviens est celle de l'uniformisation. Cependant, malgré les nombreux avantages de cette dernière, sa complexité temporelle la rend trés coûteuse pour les systèmes raides. Nous étudions ici des solutions efficaces relatives à une méthode approchée développée pour l'analyse en régime transitoire de modèles markoviens, tant des probabilités d'état que de mesures d'espérance de récompenses sur un intervalle fini. La méthode sur laquelle repose notre travail approche les probabilités d'état à l'instant t par les probabilités d'état calculées à un instant aléatoire distribué selon une loi d'Erlang. La méthode d'origine nécessite l'inversion d'une matrice (contruite à partir du générateur infinitésimal) qui fait perdre les structures spéciales telles que celles des matrices creuses ou des matrices bandes. Ce qui interdit l'usage de l'usage de la méthode d'origine pour les grands espaces d'état. Notre approche fournit des solutions efficaces qui conservent les avantages des structures particulières. Enfin nous développons des exemples qui montrent que, par rapport à la technique de l'uniformisation, notre approche est trés efficace pour les systèmes raides.

Mots-clé : Modèles markoviens, régime transitoire, uniformisation, raideur

1 Introduction

Markovian models have been widely used in the analysis of computer and communication systems. For those systems, steady state measures are good approximations of transient quantities, if the system is observed over a finite but large time interval when compared to the interval between the occurrence of events in the system. However, in many cases, it is important to obtain solutions for a relatively small time interval. Availability and performability models [1] are examples where transient analysis has proven to be useful. Transient measures have also been required during intermediate steps to obtain the steady state solution for non-Markovian models (e.g. [3, 5]).

Several methods have been proposed to obtain transient measures for Markovian models, such as those which are based on the solution of systems of partial differential equations, those which are based on time discretization, and the uniformization technique. Among these methods, the uniformization or Jensen's method [8] has become very popular in the last several years. This is due to its numerical robustness and mainly to the probabilistic interpretation which has been used in several extensions [2]. Other advantage of the method include the simplicity of implementation.

The computational cost of the uniformization is proportional to Λt, where t is the length of the observation period and Λ is a parameter which is greater than or equal to the largest absolute value of the diagonal elements of the infinitesimal generator $\mathbf{Q}$. One drawback of this method is that it requires substantial computation for stiff models. This is true since a stiff model has output rates with values that differ in orders of magnitude which, in general, implies a large value for the product Λt [6].

An approximation to obtain the transition probabilities computed over a finite interval has been proposed by Ross in [10]. The technique approximates the probability that the model is in a state at time t by the equivalent probability computed over an interval with length that has an Erlangian distribution with mean t. The method is easy to implement and from the examples presented in [10], it produces accurate results. However, the equations obtained in [10] require the inversion of a matrix which is obtained from and has the same dimension as the transition rate matrix of the model being solved. As a consequence, models with large state space cardinalities can not be solved due to the fill-in which normally occurs when the inverse is calculated.

A few extensions to the basic approach of Ross was proposed in [12]. However, it has the same computational characteristics as the original method.

In this paper we propose efficient methods to calculate transient state probabilities of Markov models, and cumulative expected reward measures over a finite interval, from the approach of [10]. We show how the measures can be computed from iterative and direct solution techniques used for solving linear equations (e.g. see [11, 4], but adapted for our specific problem in order to improve the computational efficiency of the approach. Furthermore, we point out that the solutions employed preserve special structures in the model.

Another goal of this work is to study the behavior of the proposed solutions to solve stiff models. The experiments show that the computational cost required by our method is significantly smaller than that required by the uniformization technique when used for the class of models studied, without affecting much the accuracy of the results.

In the section 2 we introduce the uniformization method and the approximation proposed by Ross in [10]. Section 3 presents the proposed iterative and direct solutions. In section 4 we evaluate the behavior of the solutions when they are used to solve stiff models. Section

5 describes additional characteristics of the method that are useful to calculate measures of interest. Our conclusions are presented in section 6.

2 Basic Concepts

2.1 The Uniformization Method

As mentioned in section 1, the uniformization method is widely used to calculate transient measures and it is considered one of the best techniques by many authors. In the method, the solution is obtained solving the problem in discrete time which provides several computational advantages.

Let $\mathcal{X} = \{X(t) : t \geq 0\}$ be a homogeneous continuous time Markov chain with a finite state space $\mathcal{S} = \{1, \ldots, M\}$, and transition rate matrix $\mathbf{Q}$. Let q_{ij} be the (i,j)th element of $\mathbf{Q}$, and $q_i = \sum_{j \neq i} q_{ij}$. Then q_i is the exponential rate out of state i. Recall that, in this process, the transition probability p^*_{ij} from state i to state j is equal to q_{ij}/q_i, and the time the chain remains in state i is exponential with mean $1/q_i$.

Let $\mathcal{Z} = \{Z_n : n = 0, 1, \ldots\}$ be a discrete time Markov chain with discrete state space $\mathcal{S}$, and transition probability matrix $\mathbf{P} = [p_{ij}]$, $\mathbf{P} = \mathbf{I} + \mathbf{Q}/\Lambda$, where $\Lambda \geq \max_i\{q_i\}$. It is not difficult to see that the transition probability from i to j given that the transition is to a state different from i (i.e., $p_{ij}/(1 - p_{ii})$) is equal to the probability p^*_{ij} of the continuous chain.

Consider now a Poisson process $\mathcal{N} = \{N(t) : t \geq 0\}$ with rate Λ independent of $\mathcal{Z}$. If the time between transitions (out of a state or back to the same state i) for the chain $\mathcal{Z}$ is exponential with rate Λ, it is possible to show that the residence time spent in a visit to i is exponential with mean $1/q_i$.

Since the total residence time in i is identical in both processes as well as the probability of moving from i to j given that a transition occurs to a state different from i, we may consider $\mathcal{X}$ and $\mathcal{Z}$ equivalent processes.

Let $\boldsymbol{\pi}(t)$ be a vector such that the jth element is equal to the probability that $\mathcal{X}$ is in state j at time t, given an initial distribution of states. After n transitions, $\mathcal{Z}$ will be in state j with probability $v_j(n)$, where $v_j(n)$ is the jth entry of the vector $\mathbf{v}(n) = \mathbf{v}(0)\mathbf{P}^n$ and $\mathbf{v}(0)$ is the initial state probability vector. Unconditioning on the number of transitions in (0,t), we obtain

$$\boldsymbol{\pi}(t) \;=\; \sum_{n=0}^{\infty} \mathrm{e}^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \mathbf{v}(n), \tag{1}$$

which is the basic equation of the uniformization method.

It is important to note that the above equation has many interesting properties. First, $\mathbf{v}(n)$ can be evaluated recursively by $\mathbf{v}(n) = \mathbf{v}(n-1)\mathbf{P}$ with $\mathbf{v}(0) = \boldsymbol{\pi}(0)$, and so if $\mathbf{P}$ is sparse, the computational cost can be reduced. Also, it is easy to obtain error bounds when the infinite series in (1) is truncated. The error bounds can be calculated from the properties of the Poisson distribution. If we truncate (1) at N, the error $\varepsilon(N)$ of any entry of vector $\boldsymbol{\pi}(t)$ is given by:

$$\varepsilon(N) \;\leq\; 1 - \sum_{n=0}^{N} \mathrm{e}^{-\Lambda t} \frac{(\Lambda t)^n}{n!}. \tag{2}$$

As a consequence, it can be shown that the computational cost is proportional to Λt. More details can be found in [2].

2.2 Ross's Approximation

Ross [10] proposed to approximate $\pi_{ij}(t)$ (the probability that $\mathcal{X}$ is in state j at time t given that the process is in state i at time 0) by the state probability calculated at a random time with Erlangian distribution.

Let $\mathcal{E}$ be an event that occurs at intervals which are independent exponential random variables Y_1, Y_2,... with rate λ independent of $\mathcal{X}$. Given that $\mathcal{X}$ is in state i, a transition in $\mathcal{X}$ occurs before event $\mathcal{E}$ with probability $q_i/(q_i+\lambda)$. Likewise, the event $\mathcal{E}$ occurs before a transition of $\mathcal{X}$ with probability $\lambda/(q_i+\lambda)$. In the first case the probability that $\mathcal{X}$ makes a transition from state i to state j is q_{ij}/q_i. In the second case, obviously process $\mathcal{X}$ remains in the same state.

Conditioning on which event occurs first, i.e. $\mathcal{E}$ or a transition of $\mathcal{X}$, and using the Markov property, we obtain $\pi_{ij}(Y_1)$ as follows:

$$\pi_{ij}(Y_1) \;=\; \frac{q_i}{q_i+\lambda}\sum_{k\neq i}\pi_{kj}(Y_1)\frac{q_{ik}}{q_i} + \frac{\lambda}{q_i+\lambda}\delta_{ij}, \tag{3}$$

where $\delta_{ij}=1$ if $i=j$ and $\delta_{ij}=0$ otherwise.

Let $Y_1, Y_2, \ldots$ be a sequence of identical and independent exponential random variables which are independent of $\mathcal{X}$. Let $Y(r) = Y_1 + \ldots + Y_r$. From equation (3) the following recursion can be obtained, recalling that $\mathcal{X}$ remains in the same state if $\mathcal{E}$ occurs before a transition of $\mathcal{X}$:

$$\pi_{ij}(Y(r)) \;=\; \frac{1}{q_i+\lambda}\sum_{k\neq i}\pi_{kj}(Y(r))q_{ik} + \frac{\lambda}{q_i+\lambda}\pi_{ij}(Y(r-1)) \tag{4}$$

where $\pi_{ij}(Y(0)) = \delta_{ij}$.

Equation (4) can be written in matrix form as:

$$-\lambda\Pi(r-1) = (\mathbf{Q}-\lambda\mathbf{I})\,\Pi(r) \tag{5}$$

where $\Pi(r) = [\pi_{ij}(Y(r))]$. Finally,

$$\Pi(r) = (\mathbf{I}-\mathbf{Q}/\lambda)^{-1}\,\Pi(r-1). \tag{6}$$

From (6) we obtain an approximation to $\Pi(t)$ in the following way. $Y(R)$ has Erlangian distribution with R stages and $E[Y(R)] = R/\lambda$. If we choose $\lambda = R/t$ then, when $R\to\infty$, the random variable $Y(R)$ approaches $E[Y(R)]$ with probability 1, and so $Y(R)$ approaches the value t. As a consequence, $\Pi(R)$ should be a good approximation to $\Pi(t)$ to large values of R.

3 Solution Approaches

In the original method of [10], the solution of equation (6) requires the inversion of matrix $(\mathbf{I}-\mathbf{Q}/\lambda)$, which can be impractical except for very small models. In this section we are concerned with alternative ways to obtain the desired solution in order to take into account special structures in the model and reduce the amount of storage and the number of operations that are needed to solve it. We study both iterative and direct solution approaches.

We first modify equation (6) as follows. Assume that the original process is uniformized as it was described in section 2. Let $\mathcal{X}^*$ be the uniformized process and $\mathbf{P}$ its transition probability matrix in which the (i,j)th element (p_{ij}) is the probability of moving from state i to state j. We recall that the time the process spends in each state before a transition occurs is exponential with rate Λ. As it was done to obtain (4), conditioning on the event that occurs first, i.e. a transition of $\mathcal{X}^*$ or the event $\mathcal{E}$, we have:

$$\pi_{ij}(Y(r)) \;=\; \frac{\Lambda}{\Lambda+\lambda}\sum_k \pi_{kj}(Y(r))p_{ik} + \frac{\lambda}{\Lambda+\lambda}\pi_{ij}(Y(r-1)). \tag{7}$$

Let $\mathbf{1}_i$ be a vector for which the ith element is equal to one and the others are zero. Let $\mathbf{z}(i,r) = \Pi(r)\mathbf{1}_i$. From (7) we obtain:

$$\left(\mathbf{I} - \frac{\Lambda}{\Lambda+\lambda}\mathbf{P}\right)\mathbf{z}(i,r) = \frac{\lambda}{\Lambda+\lambda}\mathbf{z}(i,r-1). \tag{8}$$

For $\lambda = R/t$, $\mathbf{z}(i,R)$ is an approximation of the ith column of $\Pi(t)$, that is, the j-th element of vector $\mathbf{z}(i,R)$ is an approximation for $\pi_{ji}(t)$. Equation 8 is of the type $\mathbf{Ax} = \mathbf{b}$ and we next study ways to solve it efficiently . In what follows, for notational convenience, we omit the index i of $\mathbf{z}(i,r)$ that indicates the final state considered for the transient measures being evaluated.

It is interesting to observe that we can easily obtain an approximation to the probability that the system is in a set of states $\mathcal{D}$ at time t (starting from any state), from (7). If we define $\mathbf{1}_\mathcal{D}$ as the vector for which all entries corresponding to states in set $\mathcal{D}$ are equal to one and zero otherwise, then clearly letting $\mathbf{z}(\mathcal{D},r) = \Pi(r)\mathbf{1}_\mathcal{D}$ and replacing $\mathbf{z}(i,r)$ by $\mathbf{z}(\mathcal{D},r)$ (and $\mathbf{z}(i,r-1)$ by $\mathbf{z}(\mathcal{D},r-1)$) in (8) gives the desired equation. This is true since vector $\mathbf{z}(\mathcal{D},r)$ is the sum of columns in $\Pi(r)$ corresponding to the non-zero entries in $\mathbf{1}_\mathcal{D}$.

3.1 Iterative Techniques

If the transition rate matrix of the model being studied is sparse, $\mathbf{I}-\Lambda/(\Lambda+\lambda)\mathbf{P}$ is also sparse, and iterative techniques are attractive in this case to solve (8), for example using the SOR method [4]. The final solution $\mathbf{z}(R)$ is obtained after R partial solutions for $\mathbf{z}(r), r = 1,\ldots,R$. Assume that the mean number of steps to solve (8) using an iterative method is N for any value of r. Then, the final solution requires RN vector by matrix multiplications.

The right side of equation (8) is the product of a constant by the solution in the previous step. It is known that the number of steps of an iterative method can be reduced if the initial solution is an approximation of the required solution. Therefore, if $\mathbf{z}(r-1)$ is a good approximation for $\mathbf{z}(r)$, the total number of steps of the algorithm can be reduced. The following lemma provides evidence that the approximation should be good.

Lemma 1 *The vector $\mathbf{z}(r)$ obtained at the step $r < R$ in the computation of $\boldsymbol{\pi}_i(t)$ with R steps is equal to the solution $\boldsymbol{\pi}_i(t')$ (with r steps) for $t' = (r/R)t$.*

Proof: It is trivial. The value of λ' to obtain $\boldsymbol{\pi}_i(t')$ with r steps is r/t', and so it is equal to the value of λ to obtain $\boldsymbol{\pi}_i(t)$ with R steps (that is, $\lambda' = r/t' = R/t = \lambda$, for $t' = (r/R)t$). Therefore, $\mathbf{z}(r)$ with $\lambda = R/t$ is an approximation of $\boldsymbol{\pi}_i((r/R)t)$ with r steps. □

From Lemma 1, $\mathbf{z}(r+1)$ and $\mathbf{z}(r)$ are approximations of $\boldsymbol{\pi}_i((r+1)/R)t)$ (with $r+1$ steps) and $\boldsymbol{\pi}_i((r/R)t)$ (with r steps), respectively. Therefore, it is not difficult to see that each element

of the vector $\mathbf{z}(r+1)$ is a good approximation of the corresponding element in $\mathbf{z}(r)$, for "large" values of r and R.

In Figure 1 we use the polling example of section 4 to illustrate the behavior described above. The figure shows the difference $\mathbf{z}(r+1) - \mathbf{z}(r)$ for a randomly chosen element of each vector for r varying from 1 to 30. This difference decreases from 0.072 to 0.006 as the value of r increases. (In this example, the final value of the chosen element for $r = 30$ is equal to 0.121, and the difference remains below 10% for all values of r. Furthermore, this behavior is typical for all elements.)

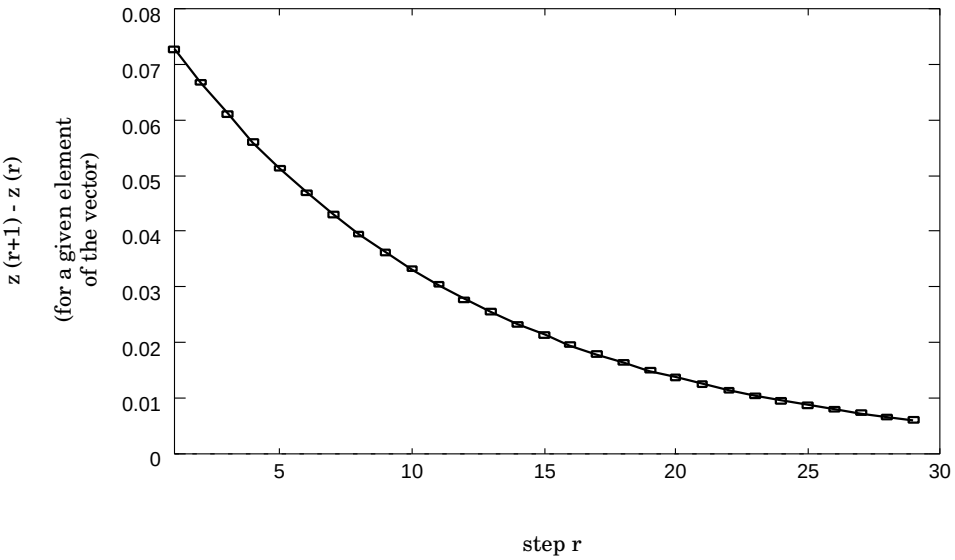

Figure 1: $\mathbf{z}_i(r+1)$ and $\mathbf{z}_i(r)$

3.2 Direct Methods

Many computer system models have transition rate matrices with block structures, for instance block Hessenberg. In what follows we propose a solution to equation (8) (for $r = 1, \ldots, R$) assuming that $\mathbf{I} - \Lambda/(\Lambda+\lambda)\mathbf{P}$ is in block form, and show the computational advantages that can be gained if one uses the reduction technique described. We assume that $\mathbf{A} = \mathbf{I} - \Lambda/(\Lambda+\lambda)\mathbf{P}$ is divided in blocks as shown below:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \ldots & \mathbf{A}_{1K} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \ldots & \mathbf{A}_{2K} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ \mathbf{A}_{K1} & \mathbf{A}_{K2} & \ldots & \mathbf{A}_{KK} \end{bmatrix} \tag{9}$$

We propose a reduction approach that requires the inversion of the diagonal blocks of $\mathbf{A}$.

Assume vector $\mathbf{z}(r)$ is partitioned into K blocks $\mathbf{z}_1(r), \mathbf{z}_2(r), \ldots \mathbf{z}_K(r)$, each one with dimension B, matching the partitioning for $\mathbf{A}$. Equation 8 can be rewritten as:

$$\sum_{j=1}^{r} \mathbf{A}_{ij}^{(0)} \mathbf{z}_j(r) = \mathbf{w}_i^{(0)}(r) \qquad i = 1, 2, ..., K. \tag{10}$$

where

$$\mathbf{A}_{ij}^{(0)} = \left(\mathbf{I} - \frac{\Lambda}{\Lambda+\lambda}\mathbf{P}_{ij}\right) \text{ and } \mathbf{w}_i^{(0)}(r) = \frac{\lambda}{\Lambda+\lambda}\mathbf{z}(r-1).$$

The superscript identifies the steps in the reduction procedure we describe below. In order to simplify the notation, we omit the index r of vectors $\mathbf{z}$ and $\mathbf{w}$.

From Corollary 1 below the diagonal block $\mathbf{A}_{11}^{(0)}$ can be inverted, and so we can solve the first equation ($i = 1$) for $\mathbf{z}_1$, and obtain:

$$\mathbf{z}_1 = -(\mathbf{A}_{11}^{(0)})^{-1} \sum_{j=2}^{K} \mathbf{A}_{ij}^{(0)}\mathbf{z}_j + (\mathbf{A}_{11}^{(0)})^{-1}\mathbf{w}_1^{(0)}. \tag{11}$$

Substituting $\mathbf{z}_1$ in (10), we obtain:

$$\sum_{j=2}^{K} \left[\mathbf{A}_{ij}^{(0)} - \mathbf{A}_{i1}^{(0)}(\mathbf{A}_{11}^{(0)})^{-1}\mathbf{A}_{1j}^{(0)}\right]\mathbf{z}_j = \left[\mathbf{w}_i^{(0)} - \mathbf{A}_{i1}^{(0)}(\mathbf{A}_{11}^{(0)})^{-1}\mathbf{w}_1^{(0)}\right] \qquad i = 2, 3, ..., K. \tag{12}$$

Equation (12) is the basis for the reduction procedure. It can be rewritten as $\mathbf{A}^{(1)}\mathbf{z}^{(1)} = \mathbf{w}^{(1)}$, where $\mathbf{z}^{(1)}$ is identical to $\mathbf{z}$ but without the first block element $\mathbf{z}_1$, $\mathbf{A}^{(1)}$ has dimension $(K-1)\times(K-1)$ blocks and $\mathbf{w}^{(1)}$ has dimensions $K-1$ blocks.

We can apply this procedure recursively for $k = 2, 3, \ldots, K-1$ since, from Corollary 1 (below), the blocks $\mathbf{A}_{kk}^{(k-1)}$ are all invertible. After $K-1$ steps, we obtain:

$$\mathbf{A}_{KK}^{(K-1)}\mathbf{z}_K = \mathbf{w}_K^{(K-1)}. \tag{13}$$

Solving the system above, we obtain $\mathbf{z}_K$ and then the other block elements of $\mathbf{z}$ can be calculated using:

$$\mathbf{z}_i = (\mathbf{A}_{ii}^{(i-1)})(-1)\left[\mathbf{w}_i^{(i-1)} - \sum_{j=0}^{(K-i-1)} \mathbf{A}_{i,K-j}^{(i-1)}\mathbf{z}_{K-j}\right]. \tag{14}$$

Figure 2 illustrates the reduction performed.

The reduction algorithm can be summarized as shown below. Note that the reduction for $\mathbf{A}$ is done only for $r = 1$.

1. Initialization step:
   $\mathbf{A}^{(0)} = (\mathbf{I} - \frac{\Lambda}{(\lambda+\Lambda)}\mathbf{P})$
   $\mathbf{w}^{(0)}(0) = \mathbf{z}(0) * \frac{\lambda}{(\lambda+\Lambda)}$
2. Reduction procedure:
   For $1 \le k \le K-1$
   - $\mathbf{A}_{ij}^{(k)} = \mathbf{A}_{ij}^{(k-1)} - \mathbf{A}_{ik}^{(k-1)}(\mathbf{A}_{kk}^{(k-1)})^{-1}\mathbf{A}_{kj}^{(k-1)} \qquad k+1 \le i, j \le K,$

The vectors $\mathbf{z}(r)$ for $1 \le r \le R$ can now be calculated as follows:

- For $1 \le r \le R$
  - for $1 \le k \le K-1$
    $\mathbf{w}_i^{(k)}(r-1) = \mathbf{w}_i^{(k-1)}(r-1) - \mathbf{A}_{ik}^{(k-1)}(\mathbf{A}_{kk}^{(k-1)})^{-1}\mathbf{w}_k^{(k-1)}(r-1)$
    $k+1 \le i \le K$

$$\begin{bmatrix} A_{11}^{(0)} & A_{12}^{(0)} & A_{13}^{(0)} & & \cdots & & A_{1K}^{(0)} \\ A_{21}^{(0)} & A_{22}^{(1)} & A_{23}^{(1)} & A_{24}^{(1)} & \cdots & & A_{2K}^{(1)} \\ A_{31}^{(0)} & A_{32}^{(1)} & A_{33}^{(2)} & A_{34}^{(2)} & \cdots & & A_{3K}^{(2)} \\ & A_{42}^{(1)} & A_{43}^{(2)} & A_{44}^{(3)} & A_{45}^{(3)} & \cdots & A_{4K}^{(3)} \\ \cdot & & \cdot & & \cdot & \cdot & \cdot \\ \cdot & & \cdot & & \cdot & \cdot & \cdot \\ \cdot & & \cdot & & & \cdot & \cdot \\ A_{K1}^{(0)} & A_{K2}^{(1)} & A_{K3}^{(2)} & A_{K4}^{(3)} & \cdots & & A_{KK}^{(k-1)} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ \cdot \\ \cdot \\ \cdot \\ z_K \end{bmatrix} = \begin{bmatrix} w_1^{(0)} \\ w_2^{(1)} \\ w_3^{(2)} \\ w_4^{(3)} \\ \cdot \\ \cdot \\ \cdot \\ w_K^{(K-1)} \end{bmatrix}$$

Figure 2: Final matrix and vector $\mathbf{w}$ after the reduction

– Calculate $\mathbf{z}(r)$ and $\mathbf{w}^{(0)}(r)$
  * $\mathbf{z}_i(r) = (\mathbf{A}_{ii}^{(i-1)})^{-1} \left[\mathbf{w}_i^{(i-1)}(r-1) - \sum_{j=0}^{K-i-1} \mathbf{A}_{i,K-j}^{(i-1)} \mathbf{z}_{K-j}(r)\right]$
    $1 \leq i \leq K$
  * $\mathbf{w}^{(0)}(r) = \mathbf{z}(r)\frac{\lambda}{(\lambda+\Lambda)}$

In what follows we shown that the diagonal blocks obtained in the reduction procedure are all invertible. In fact, all blocks $\mathbf{A}_{ij}^{(k)}$ have a special form as it will be shown next.

Definition 1 *A matrix* $\mathbf{M}$ *will be said to be of the form* $\mathcal{F}$ *if* $\mathbf{M}$ *can be written as* $(\mathbf{I}-\mathbf{S})$ *where* $\mathbf{S}$ *is a substochastic matrix.*

Note that the matrix $\mathbf{A}^{(0)} = \mathbf{I} - c\mathbf{P}$ where $c = \frac{\Lambda}{\Lambda+\lambda} < 1$ is of this form. Moreover each diagonal block $\mathbf{A}_{ii}^{(0)}$ is also of the form $\mathcal{F}$ and each off diagonal block $\mathbf{A}_{ij}^{(0)}$ is such that $-\mathbf{A}_{ij}^{(0)}$ is substochastic.

Theorem 1 *Step 2 of the algorithm preserves the form* $\mathcal{F}$ *of* $\mathbf{A}^{(0)}$*, that is:*

a) $\forall k, \forall i > k$, $\mathbf{A}_{ii}^{(k)}$ *is of the form* $\mathcal{F}$*;*

b) $\forall k, \forall i, j > k$, $j \neq i$, $-\mathbf{A}_{ij}^{(k)}$ *is substochastic.*

Proof:

1. For the case $k = 1$ of the reduction procedure :

a) For the diagonal blocks:

$$\mathbf{A}_{jj}^{(1)} = \mathbf{A}_{jj}^{(0)} - \mathbf{A}_{j1}^{(0)}[\mathbf{A}_{11}^{(0)}]^{-1}\mathbf{A}_{1j}^{(0)}, \qquad 2 \leq j \leq K \tag{15}$$

But

$$\begin{aligned} \mathbf{A}_{jj}^{(0)} &= \mathbf{I} - c\mathbf{P}_{jj} \\ \mathbf{A}_{j1}^{(0)} &= -c\mathbf{P}_{j1} \\ \mathbf{A}_{1j}^{(0)} &= -c\mathbf{P}_{1j} \\ \mathbf{A}_{11}^{(0)} &= \mathbf{I} - c\mathbf{P}_{11} \end{aligned}$$

where $c = \frac{\Lambda}{\Lambda+\lambda} < 1$ and $\mathbf{P}_{ij}$ is the $(i,j)^{th}$ block of the stochastic matrix $\mathbf{P}$.

Therefore,

$$\begin{aligned}\mathbf{A}_{jj}^{(1)} &= \mathbf{I} - c\mathbf{P}_{jj} - c\mathbf{P}_{j1}[\mathbf{I} - c\mathbf{P}_{11}]^{-1}c\mathbf{P}_{1j} \\ &= \mathbf{I} - c\left[\mathbf{P}_{jj} + c\mathbf{P}_{j1}[\mathbf{I} - c\mathbf{P}_{11}]^{-1}\mathbf{P}_{1j}\right]\end{aligned}$$

The matrix $c\mathbf{P}_{11}$ being substochastic the inverse $[\mathbf{I} - c\mathbf{P}_{11}]^{-1}$ exists and is such that :

$$[\mathbf{I} - c\mathbf{P}_{11}]^{-1} = \sum_{n=0}^{\infty} c^n \mathbf{P}_{11}^n$$

The block $\mathbf{P}_{11}$ is itself a substochastic matrix and the supremum of each element of $c\mathbf{P}_{j1}[\mathbf{I} - c\mathbf{P}_{11}]^{-1}\mathbf{P}_{1j}$ is obtained for $c = 1$, as a consequence :

$$\mathbf{P}_{jj} + c\mathbf{P}_{j1}[\mathbf{I} - c\mathbf{P}_{11}]^{-1}\mathbf{P}_{1j} < \mathbf{P}_{jj} + \mathbf{P}_{j1}[\mathbf{I} - \mathbf{P}_{11}]^{-1}\mathbf{P}_{1j}$$

where the inequality refers to each element of the two matrices.

But $\widehat{\mathbf{P}}_{jj}^{(1)} = \mathbf{P}_{jj} + \mathbf{P}_{j1}[\mathbf{I} - \mathbf{P}_{11}]^{-1}\mathbf{P}_{1j}$ is the j^{th} diagonal block element of the stochastic complement for matrix $\mathbf{P}$ with respect to $\mathbf{P}_{11}$, so it is a substochastic matrix.

Then

$$\mathbf{A}_{jj}^{(1)} = \mathbf{I} - c\widetilde{\mathbf{P}}_{jj}^{(1)}$$

where $\widetilde{\mathbf{P}}_{jj}^{(1)} < \widehat{\mathbf{P}}_{jj}^{(1)}$ and so $\widetilde{\mathbf{P}}_{jj}^{(1)}$ is a substochastic matrix. Therefore $\mathbf{A}_{jj}^{(1)}$ is of the form $\mathcal{F}$.

b) For the off diagonal blocks, the proof is similar :

$$\mathbf{A}_{ij}^{(1)} = \mathbf{A}_{ij}^{(0)} - \mathbf{A}_{i1}^{(0)}[\mathbf{A}_{11}^{(0)}]^{-1}\mathbf{A}_{1j}^{(0)}, \qquad 2 \le j \le K \tag{16}$$

But $\mathbf{A}_{ij}^{(0)} = -c\mathbf{P}_{ij}$ and therefore

$$\begin{aligned}\mathbf{A}_{ij}^{(1)} &= -c\mathbf{P}_{ij} - c\mathbf{P}_{i1}[\mathbf{I} - c\mathbf{P}_{11}]^{-1}c\mathbf{P}_{1j} \\ &= -c\left[\mathbf{P}_{ij} + c\mathbf{P}_{i1}[\mathbf{I} - c\mathbf{P}_{11}]^{-1}\mathbf{P}_{1j}\right]\end{aligned}$$

For the same reason as in the previous case, we have :

$$\mathbf{P}_{ij} + c\mathbf{P}_{i1}[\mathbf{I} - c\mathbf{P}_{11}]^{-1}\mathbf{P}_{1j} < \mathbf{P}_{ij} + \mathbf{P}_{i1}[\mathbf{I} - \mathbf{P}_{11}]^{-1}\mathbf{P}_{1j}$$

And $\widehat{\mathbf{P}}_{ij}^{(1)} = \mathbf{P}_{ij} + \mathbf{P}_{i1}[\mathbf{I} - \mathbf{P}_{11}]^{-1}\mathbf{P}_{1j}$ being the $(i,j)^{th}$ block element of the stochastic complement for matrix $\mathbf{P}$ with respect to $\mathbf{P}_{11}$ is a substochastic matrix.

Then $\mathbf{A}_{ij}^{(1)} = -c\widetilde{\mathbf{P}}_{ij}^{(1)}$ where $\widetilde{\mathbf{P}}_{ij}^{(1)} < \widehat{\mathbf{P}}_{ij}^{(1)}$ and so $\widetilde{\mathbf{P}}_{ij}^{(1)}$ is substochastic.

2. For $k = 2$:

So far have proved above that after the first step of the reduction procedure, $\mathbf{A}^{(1)}$ is of the form $\mathcal{F}$ since each diagonal block $\mathbf{A}_{jj}^{(1)}$ can be written as $\mathbf{A}_{jj}^{(1)} = \mathbf{I} - c\widetilde{\mathbf{P}}_{jj}^{(1)}$ and each off diagonal

block $\mathbf{A}_{ij}^{(1)}$ can be written as $\mathbf{A}_{ij}^{(1)} = -c\tilde{\mathbf{P}}_{ij}^{(1)}$ with $\tilde{\mathbf{P}}_{ij}^{(1)} < \hat{\mathbf{P}}_{ij}^{(1)}$, the matrix $\hat{\mathbf{P}}^{(1)}$ being a stochastic matrix. Again we consider the two cases :

a)For the diagonal blocks:

$$\mathbf{A}_{jj}^{(2)} = \mathbf{A}_{jj}^{(1)} - \mathbf{A}_{j2}^{(1)}[\mathbf{A}_{22}^{(1)}]^{-1}\mathbf{A}_{2j}^{(1)}, \qquad 3 \leq j \leq K \tag{17}$$

Therefore,

$$\begin{aligned} \mathbf{A}_{jj}^{(2)} &= \mathbf{I} - c\tilde{\mathbf{P}}_{jj}^{(1)} - c\tilde{\mathbf{P}}_{j2}^{(1)}[\mathbf{I} - c\tilde{\mathbf{P}}_{22}^{(1)}]^{-1}c\tilde{\mathbf{P}}_{2j}^{(1)} \\ &= \mathbf{I} - c\left[\tilde{\mathbf{P}}_{jj}^{(1)} + c\tilde{\mathbf{P}}_{j2}^{(1)}[\mathbf{I} - c\tilde{\mathbf{P}}_{22}^{(1)}]^{-1}\tilde{\mathbf{P}}_{2j}^{(1)}\right] \end{aligned}$$

The matrix $c\tilde{\mathbf{P}}_{22}^{(1)}$ being substochastic the inverse $[\mathbf{I} - c\tilde{\mathbf{P}}_{22}^{(1)}]^{-1}$ exists and is such that :

$$[\mathbf{I} - c\tilde{\mathbf{P}}_{22}^{(1)}]^{-1} = \sum_{n=0}^{\infty} c^n(\tilde{\mathbf{P}}_{22}^{(1)})^n$$

For the same reason as before :

$$\tilde{\mathbf{P}}_{jj}^{(1)} + c\tilde{\mathbf{P}}_{j2}^{(1)}[\mathbf{I} - c\tilde{\mathbf{P}}_{22}^{(1)}]^{-1}\tilde{\mathbf{P}}_{2j}^{(1)} < \tilde{\mathbf{P}}_{jj}^{(1)} + \tilde{\mathbf{P}}_{j2}^{(1)}[\mathbf{I} - \tilde{\mathbf{P}}_{22}^{(1)}]^{-1}\tilde{\mathbf{P}}_{2j}^{(1)} < \hat{\mathbf{P}}_{jj}^{(1)} + \hat{\mathbf{P}}_{j2}^{(1)}[\mathbf{I} - \hat{\mathbf{P}}_{22}^{(1)}]^{-1}\hat{\mathbf{P}}_{2j}^{(1)}$$

But $\hat{\mathbf{P}}_{jj}^{(2)} = \hat{\mathbf{P}}_{jj}^{(1)} + \hat{\mathbf{P}}_{j2}^{(1)}[\mathbf{I} - \hat{\mathbf{P}}_{22}^{(1)}]^{-1}\hat{\mathbf{P}}_{2j}^{(1)}$ is the j^{th} diagonal block element of the stochastic complement for matrix $\hat{\mathbf{P}}^{(1)}$ with respect to $\hat{\mathbf{P}}_{22}^{(1)}$ (or equivalently the stochastic complement for matrix $\mathbf{P}$ with respect to $\mathbf{P}_{11}$ and $\mathbf{P}_{22}$), so it is a substochastic matrix.

Then

$$\mathbf{A}_{jj}^{(2)} = \mathbf{I} - c\tilde{\mathbf{P}}_{jj}^{(2)}$$

where $\tilde{\mathbf{P}}_{jj}^{(2)} < \hat{\mathbf{P}}_{jj}^{(2)}$ and so $\tilde{\mathbf{P}}_{jj}^{(2)}$ is a substochastic matrix. Therefore $\mathbf{A}_{jj}^{(2)}$ is of the form $\mathcal{F}$.

b)For the off diagonal blocks, the proof is similar :

$$\mathbf{A}_{ij}^{(2)} = \mathbf{A}_{ij}^{(1)} - \mathbf{A}_{i2}^{(1)}[\mathbf{A}_{22}^{(1)}]^{-1}\mathbf{A}_{2j}^{(1)}, \qquad 3 \leq j \leq K \tag{18}$$

But $\mathbf{A}_{ij}^{(1)} = -c\tilde{\mathbf{P}}_{ij}^{(1)}$ and therefore, as in the previous case, we have :

$$\begin{aligned} \mathbf{A}_{ij}^{(2)} &= -c\tilde{\mathbf{P}}_{ij}^{(1)} - c\tilde{\mathbf{P}}_{i2}^{(1)}[\mathbf{I} - c\tilde{\mathbf{P}}_{22}^{(1)}]^{-1}c\tilde{\mathbf{P}}_{2j}^{(1)} \\ &= -c\left[\tilde{\mathbf{P}}_{ij}^{(1)} + c\tilde{\mathbf{P}}_{i2}^{(1)}[\mathbf{I} - c\tilde{\mathbf{P}}_{22}^{(1)}]^{-1}\tilde{\mathbf{P}}_{2j}^{(1)}\right] \end{aligned}$$

Then

$$\tilde{\mathbf{P}}_{ij}^{(1)} + c\tilde{\mathbf{P}}_{i2}^{(1)}[\mathbf{I} - c\tilde{\mathbf{P}}_{22}^{(1)}]^{-1}\tilde{\mathbf{P}}_{2j}^{(1)} < \tilde{\mathbf{P}}_{ij}^{(1)} + \tilde{\mathbf{P}}_{i2}^{(1)}[\mathbf{I} - \tilde{\mathbf{P}}_{22}^{(1)}]^{-1}\tilde{\mathbf{P}}_{2j}^{(1)} < \hat{\mathbf{P}}_{ij}^{(1)} + \hat{\mathbf{P}}_{i2}^{(1)}[\mathbf{I} - \hat{\mathbf{P}}_{22}^{(1)}]^{-1}\hat{\mathbf{P}}_{2j}^{(1)}$$

Denoting $\hat{\mathbf{P}}_{ij}^{(2)} = \hat{\mathbf{P}}_{ij}^{(1)} + \hat{\mathbf{P}}_{i2}^{(1)}[\mathbf{I} - \hat{\mathbf{P}}_{22}^{(1)}]^{-1}\hat{\mathbf{P}}_{2j}^{(1)}$ the $(i,j)^{th}$ block element of the stochastic complement for matrix $\hat{\mathbf{P}}^{(1)}$ with respect to $\hat{\mathbf{P}}_{22}^{(1)}$ is a substochastic matrix, so we get :

$$\mathbf{A}_{ij}^{(2)} = -c\tilde{\mathbf{P}}_{ij}^{(2)}$$

where $\tilde{\mathbf{P}}_{ij}^{(2)} < \hat{\mathbf{P}}_{ij}^{(2)}$ and so $\tilde{\mathbf{P}}_{ij}^{(2)}$ is a substochastic matrix.

3. For $k = 3, ..., K$.

By applying the same arguments and noting that

$$\widetilde{\mathbf{P}}_{ij}^{(k-1)} + c\widetilde{\mathbf{P}}_{ik}^{(k-1)}[\mathbf{I} - c\widetilde{\mathbf{P}}_{kk}^{(k-1)}]^{-1}\widetilde{\mathbf{P}}_{kj}^{(k-1)} < \widehat{\mathbf{P}}_{ij}^{(k-1)} + \widehat{\mathbf{P}}_{ik}^{(k-1)}[\mathbf{I} - \widehat{\mathbf{P}}_{kk}^{(k-1)}]^{-1}\widehat{\mathbf{P}}_{kj}^{(k-1)}$$

where $\widehat{\mathbf{P}}^{(k)}$ is the stochastic complement for matrix $\widehat{\mathbf{P}}^{(k-1)}$ with respect to $\widehat{\mathbf{P}}_{kk}^{(k-1)}$, the theorem follows. □

From Theorem 1 we can see that, for $i \neq j$, the matrix operations in step 2 of the algorithm above are done using positive numbers. Likewise, the $\mathbf{w}_i^{(k)}(r)$'s have all positive entries that are less or equal than one, and the equation used to obtain these vectors involve only positive quantities. The same observation is valid for the $\mathbf{z}_i(r)$'s.

We next show that the $\mathbf{A}_{ii}^{(k)}$ are all invertible.

Corollary 1 *The diagonal blocks in the reduction procedure are all invertible.*

Proof: The result immediately follows since the $\mathbf{A}_{ii}^{(k)}$ have all the form $\mathcal{F}$. □

In order to access the computational requirements of the algorithm we assume as above, without loss of generality, that the blocks in (9) have identical size $B \times B$ and that the full matrix has K blocks. Figure 3 illustrates the algorithm.

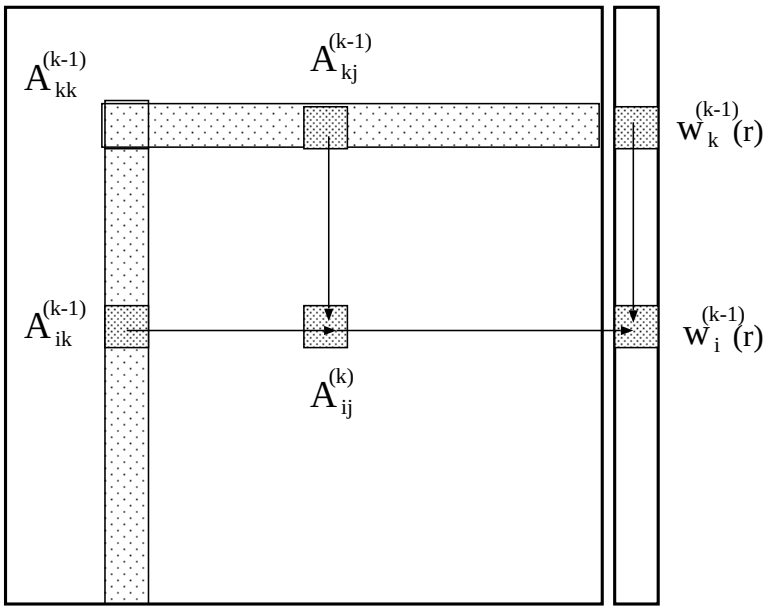

Figure 3: The algorithm

The algorithm requires the computation of the values $\mathbf{A}_{ij}^{(k)}$ in step 2. This requires $O((B \times K)^3)$ operations if the original matrix is full and has no special structure. However, one should note that the computational steps are the same as those to solve a stochastic matrix by Gaussian elimination based methods, like the GTH [7] algorithm. Furthermore, the computational savings for special structures are the same as for GTH. For instance, banded structures are preserved and only one matrix of dimension identical to the original rate matrix is necessary to store the $\mathbf{A}_{ij}^{(k)}$ block elements. Note that each step of the reduction procedure requires the

inversion of a block in the diagonal of the matrix been reduced, and from Corollary 1 this inverse always exists.

The calculations to obtain the values of $\mathbf{w}_k^{(k-1)}$ for a given value of r require $K(K+1)/2$ operations of a vector of size B by a $B \times B$ matrix which has already been obtained during the reduction process. Furthermore, only a block vector of size K $(B \times B)$ is needed for storing the $\mathbf{w}_k^{(k-1)}$. It is important to note that the reduction procedure for matrix $\mathbf{A}$ is done only once for all values of r. Then similar calculations are done but on vector $\mathbf{w}(r)$ only, for each value of r.

4 Numerical Examples

In this section we present four examples to evaluate the computational cost and precision of the solution approaches we proposed. The first two examples are stress cases, and the last two are examples of models in the literature. For the iterative technique we choose the SOR method using the initial vectors for the iteration as proposed in section 3.1.

In order to evaluate the precision of the method for stiff models we calculate the maximum relative error, the minimum relative error and the relative sum of errors. The maximum and the minimum relative errors are computed from $|\pi_i(t) - \widetilde{\pi_i}(t)|/\pi_i(t)$. The relative sum of errors is defined as $\sum_i |\pi_i(t) - \widetilde{\pi_i}(t)|$. The π_i's are the exact values obtained for the M/M/1/k queue example. For the multiplexor example, we use as exact values the results obtained with the uniformization technique. The $\widetilde{\pi}_i(t)$'s are the values obtained from the proposed solution methods.

The computational costs were evaluated in the following way. For the iterative solution, we let N be the total number of steps. and n_i be the number of iterations for step i. Recall that we use SOR for calculating the solution at each step, and so each iteration has a cost equal to a vector by matrix multiplication with vector dimension equal to the number of states in the model. Clearly, the total computational cost of the proposed method is equal to $\sum_i n_i N$ vector by matrix multiplications. The cost of the direct method is mentioned in section 3.2, and we use number of multiplications in our studies. For comparison purposes we recall that each iteration of the uniformization technique has a cost equal to a vector by matrix multiplication similar to that of the iterative technique. Furthermore, the number of iterations is proportional to Λt.

The first and the second examples we choose are not taken from any real system model, but they have a structure that usually presents numerical problems. The third example is the M/M/1/k queue model for which an explicit analytical solution is available for the transient probabilities at time t. This allow us to compare the precision of the results obtained with the proposed approaches with an exact solution. The last example is a stiff model of a multiplexor system with exponential timeouts. The main goal of this example is to compare the computational cost required by the proposed methods with the uniformization technique.

4.1 First example

The Markov chain for the first example is shown in the Figure 4. As it can be seen from the rates in the figure, the model is stiff and the transition rates differ roughly by four orders of magnitude. The uniformization rate Λ is 20.000, and Λt has a large value even if $t = 1$ as in the case we study.

A particular feature of the model is that it has a by-modal probability mass function. In other words, if the model is in state 1 (or 4), it remains in this state for a long period of time, if compared to the time it remains in states 2 or 3. Some real system models exhibit this behavior (e.g. the Aloha protocol).

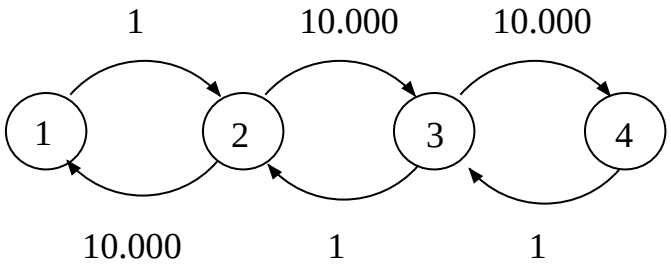

Figure 4: Markov chain for the first example

For the experiments performed, we choose 1 as the initial state. The solution of this model using the uniformization technique requires tens of thousands steps to obtain a precision of five decimal digits. Even relaxing the precision to two decimal digits, the number of steps required does not decrease by much.

If we solve the model using the proposed solution, the number of vector by matrix multiplications required is in the order of a few hundreds for $n = 20$ steps and it increases by 30% for $n = 30$ steps. (Recall that the matrix used has the same dimension and structure of the original transition rate matrix of the model.) Thus, the number of operations when the proposed solution is used is roughly two orders of magnitude smaller than the number of operations required by the uniformization technique. It is important to note that the precision of the proposed solution is two decimal digits when compared with uniformization.

In order to illustrate the reduction in the computational cost obtained as a consequence of Lemma 1, we compare the number of steps of the approach with that when we initialize the SOR method by choosing an initial solution vector with identical elements. For this example, the number of steps was reduced by half when we used the proposed solution.

The improvement obtained is model dependent and may be substantial for large models. As expected, the total computational cost is always reduced when our approach is used.

4.2 Cyclic Chain

The second example is the cyclic Markov chain shown in Figure 5. Some solution techniques have problems with this type of chains and also the model is stiff.

The computational cost when the uniformization technique is used is roughly one hundred thousand steps, for achieving a precision of five or even two decimals digits. The computational cost of the proposed iterative solution is roughly one hundred steps for $n = 10$ and 179 steps for $n = 20$ where a precision of two decimal digits is achieved when compared with the uniformization technique. Note that the cost of the proposed approach is three orders of magnitude smaller than that when uniformization is used.

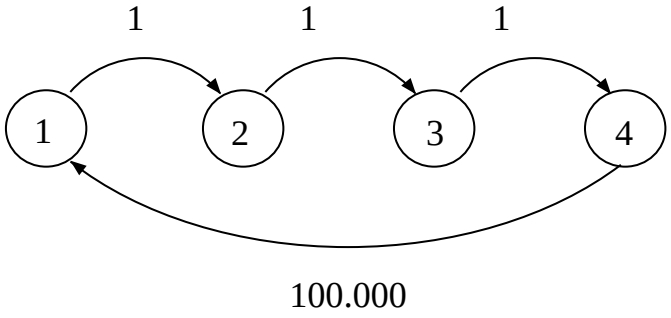

Figure 5: Cyclic Markov chain

4.3 M/M/1/K Queue

This example has a closed form solution for the transient state probabilities. The number of states in this model is $K+1$ and the parameters chosen are $\lambda = 90$, $\mu = 100$, $K = 10$ and $t = 0.01$ units of time. The parameter t is chosen such that the steady state is not reached for this model.

The exact values for the state probabilities at time t can be computed from [9]:

$$\pi_{ij}(t) = \Pi_j + \frac{2\rho^{(j-i)/2}}{k+1} \sum_{l=1}^{k} \frac{1}{x_l} \left[\sin\left(\frac{li\pi}{k+1}\right) - \sqrt{\rho}\, \sin\left(\frac{l(i+1)\pi}{k+1}\right) \right] \tag{19}$$

$$\left[\sin\left(\frac{lj\pi}{k+1}\right) - \sqrt{\rho}\, \sin\left(\frac{l(j+1)\pi}{k+1}\right) \right] e^{-\gamma_l t} \tag{20}$$

where i is the initial state, j is the state at time t, $\rho = \lambda/\mu$ and Π_j is the steady state probability for state j. Furthermore the x_l's are calculated from the eigenvalues γ_l obtained from (21).

$$\gamma_l = \lambda + \mu - 2\sqrt{\lambda\mu}\, cos\left(\frac{l\pi}{k+1}\right) = \mu x_l, \qquad l = 1, 2, ..., k \tag{21}$$

We have performed $K+1 = 11$ experiments and for each a different initial state was used. The number of steps considered was $R = 30$. The values of the errors presented in the table 1 have been obtained from the values of $\pi_i(0.01)$ and $\widetilde{\pi}_i(0.01)$ computed for each one of the experiments. We note that the precision obtained is very good.

4.4 A Model of a Multiplexor Channel

We now consider a multiplexor channel modeled by a polling system consisting of two packet queues q_1 and q_2 which are visited by a single server in cyclic order. The queue q_i $(i = 1, 2)$ is served until it is emptied or a timeout $temp_i$ expires. If the two queues are empty, the server waits until the arrival of a packet and then immediately begins serving the queue which stores the packet that has just arrived. The switching over time between queue i and $i+1$ is tc_i. We assume that all service times, interarrival times, timeouts and switchover times are exponentially distributed.

In order to keep the example small to better explain the solution methods the buffer size was chosen equal to one and we obtain a Markov chain with 11 states. The other parameters chosen for this example are as follows. The length of the packets that arrive at queue q_1 (q_2) is equal

Initial state	Max. Rel. Error	Min. Rel. Error	Sum Rel. of Errors
0	5.62e-03	4.73e-08	1.29e-02
1	7.98e-03	3.05e-07	1.69e-02
2	2.69e-03	1.98e-06	1.74e-02
3	8.11e-03	1.11e-05	1.76e-02
4	8.08e-03	5.15e-05	1.84e-02
5	8.08e-03	1.40e-04	1.86e-02
6	8.08e-03	9.70e-05	1.85e-02
7	8.10e-03	2.38e-05	1.79e-02
8	8.25e-03	4.70e-06	1.80e-02
9	7.87e-03	8.02e-07	1.73e-02
10	6.24e-03	1.35e-07	1.48e-02

Table 1: M/M/1/10 - Relative Errors

R	N. of steps	Max. Rel. Error	Min. Rel. Error	Sum Rel. of Errors
10	144	2.33e-02	1.73e-08	5.03e-02
20	246	1.17e-02	9.05e-09	2.56e-02
30	331	7.79e-03	6.13e-09	1.72e-02

Table 2: Multiplexor model - Relative error

to 400 (600) bits. The server models the multiplexor communication channel and its capacity is equal to $1.5Mb/s$. The input load for queue q_1 is equal to 10% of the channel capacity and for queue q_2 it is 60%. The values of the timeouts are $temp_1 = 2ms$ and $temp_2 = 8ms$. The switchover times are $tc_1 = tc_2 = 0.1\mu sec$. (Note that the switchover times are very small and the resulting chain is stiff.) The observation period is $t = 0.5$.

In the first set of experiments, we use the proposed iterative solution method to access the computational gains to solve the (stiff) model when compared with uniformization. For the comparison, the final state chosen was that where the number of packets in the q_1 and q_2 buffers is 1 and 0, respectively, and the server is serving q_1. To evaluate the precision of the method, we vary the number of steps from $R = 10$ until $R = 30$.

Table 2 shows the relative errors obtained for each value of R and the computational cost in terms of the number of vector by matrix multiplications performed. From the table we can observe that a precision of two decimals places was obtained even when $R = 10$. Furthermore, the computational cost of the iterative technique is much smaller than that when uniformization is used. In this example, the solution using uniformization required $5,340$ iterations.

In the second set of experiments, we solve the model by the reduction procedure we propose. The final state chosen is the same as for the first set of experiments.

In table 3 the state probabilities and the maximum relative errors obtained for each value of R are presented. We can observe that a precision of two decimals places is obtained for all values of R as when the iterative solution is used.

We evaluate the computational cost in terms of the number of multiplications performed. Table 4 shows the computational cost of the uniformization, the iterative solution and the

Initial state	Uniformization	R=10	R=20	R=30
1	2.8693e-01	3.0201e-01	2.9432e-01	2.9179e-01
2	1.4230e-01	1.3601e-01	1.3895e-01	1.4000e-01
3	2.8691e-01	2.8706e-01	2.8722e-01	2.8715e-01
4	2.0997e-01	1.9911e-01	2.0425e-01	2.0606e-01
5	2.8697e-01	3.0204e-01	2.9435e-01	2.9182e-01
6	2.9236e-01	2.8661e-01	2.8948e-01	2.9040e-01
7	2.0999e-01	1.9913e-01	2.0427e-01	2.0608e-01
8	2.9236e-01	2.8661e-01	2.8948e-01	2.9040e-01
9	2.8695e-01	3.0202e-01	2.9433e-01	2.9179e-01
10	2.0995e-01	1.9908e-01	2.0422e-01	2.0604e-01
11	1.4227e-01	1.3598e-01	1.3892e-01	1.3997e-01
Max. Rel. Error		5.2523e-02	2.5733e-02	1.6913e-02

Table 3: Multiplexor Model - State Probabilities

N. of Steps	**Iter. Solution**	**Red. Procedure**	**Uniformization**
10	4752	1950	176220
20	8118	3630	
30	10923	5310	

Table 4: Multiplexor model - Computational Cost

reduction procedure for each value of R. We can observe that the reduced procedure presents the lowest computational cost. Note however, that for this experiment we are mainly concerned with the efficiency of the solution approach to solve stiff models, without taking advantage of any special structure that could speed up the solution. Note that the choice between the iterative approach and the direct procedure depends on the structure of the original transition rate matrix of the model being solved.

5 Additional Considerations

In this section we briefly describe how can expected reward measures can be calculated and show a few characteristics of the approximation described in previous sections. Assume that we associate to state i in the model a reward rate r_i that is gained each time unit the model spends in state i. Without lack of generality, assume that all rewards are positive and less or equal than one.

Let $c_{ij}(t)$ be the cumulative expected reward gained in state j during $(0,t)$ when the process started in state i. Using the same arguments that lead to equation (7), and recalling that the expected interval between the occurrence of an event in the model, event $\mathcal{E}$ or a transition from the uniformized chain, is $1/(\Lambda+\lambda)$, the expected reward accumulated during an observation period can be obtained by the recursion:

$$c_{ij}(Y(r)) = \frac{r_i}{\Lambda+\lambda}\delta_{ij} + \frac{\Lambda}{\Lambda+\lambda}\sum_{k\neq i} c_{kj}(Y(r))p_{ik} + \frac{\lambda}{\Lambda+\lambda}c_{ij}(Y(r-1)). \qquad (22)$$

where $\delta_{ij} = 1$ for $i = j$ and 0 otherwise.

Let $\mathbf{c}(i,r) = \mathbf{C}(r)\mathbf{1}_i$ where $\mathbf{C}(r) = [c_{ij}]$. Then, from (22) we obtain:

$$\left(\mathbf{I} - \frac{\Lambda}{\Lambda+\lambda}\mathbf{P}\right)\mathbf{c}(i,r) = \frac{1}{\Lambda+\lambda}\mathbf{R}\mathbf{1}_i + \frac{\lambda}{\Lambda+\lambda}\mathbf{c}(i,r-1). \tag{23}$$

where $\mathbf{R}$ is a diagonal matrix in which the i-th diagonal element is equal to r_i.

It is interesting to note that equation (23) is similar in form to equation (8) and the same algorithm to calculate approximately the probability distributions at time t works in this case. Furthermore, we can observe that, if the reduction algorithm is applied to solve (8) then not much extra work needs to be performed to solve (23), since matrix $\mathbf{A}$ is the same in both calculations and only the values of the new $\mathbf{w}^{(k)}(r)$'s and $\mathbf{c}(r)$'s need to be obtained. In addition, clearly by using $\mathbf{1}_{\mathcal{D}}$ as before, we can obtain the accumulated reward in subset of states $\mathcal{D}$.

In several cases, the measures of interest must be computed for different instants during the observation period. The uniformization procedure is particular useful for that and if one calculates the state probabilities at time t, little additional effort is done to calculate these probabilities at any time $t' < t$ (e.g. [2]). If the approximation we describe is used, we can also obtain savings if we are interested in measures at different time instants. Lemma 1 shows that intermediate steps in the solution are solutions for an instant $t' < t$. However, one should note that the solution at t' is less accurate than the one for time t, since it is a solution for step $r < R$.

Finally, suppose that we have calculated $\mathbf{z}(i,R)$ (or $\mathbf{c}(i,R)$) and one is interested in $\mathbf{z}(j,R)$ for $i \neq j$. If the direct method is used, only little additional effort is necessary, once matrix $\mathbf{A}$ is reduced. Note that the only additional calculations necessary are those to obtain the $\mathbf{w}^{(k)}(r)$'s and the $\mathbf{z}(r)$'s, which can be performed at a cost significantly smaller than that required for reducing $\mathbf{A}$.

6 Conclusions

In this paper we have proposed efficient solutions to the basic method of [10] to obtain the state probabilities at time t. The solutions we propose can take advantage of special structures in the transition rate matrix of the model of interest, which is essential for solving large models. Furthermore, expected cumulative reward measures can also be calculated from the same basic algorithm.

Our study in this work was mainly focused on the behavior of the method for stiff models. We have only presented a subset of the examples we run to illustrate the efficacy of the proposed approaches. The examples we present show that the accuracy achieved is satisfactory, and large computational gains can be obtained even for small models, when the approach is compared with uniformization.

Unlike uniformization, the method we use do not provide any easy computable error bounds. However, our studies show that good accuracy is achieved even for a small number of steps, in all models we solved. Furthermore, the precision seems not to be affected by the degree of stiffness in the model.

References

[1] E. de Souza e Silva and H.R. Gail. Performability analysis of computer systems: from model specification to solution. *Performance Evaluation*, 14:157–196, 1992.

[2] E. de Souza e Silva and H.R. Gail. The uniformization method in performability analysis. In R. Marie, B. Haverkort, G. Rubino, Nico Van Dijk, and K. Trivedi, editors, *Second International Worshop on Performability Modeling of Computer and Communication Systems*, 1993.

[3] E. de Souza e Silva, H.R. Gail, and R.R. Muntz. Polling systems with server timeouts and their application to token passing networks. *IEEE/ACM Transactions on Networking*, 3(5):560–575, 1995.

[4] G.H. Golub and C.F. van Loan. *Matrix Computation, 2nd edition.* The John Hopkins University Press, 1989.

[5] W.K. Grassmann. The GI/PH/1 queue: a method to find the transition matrix. *INFOR*, 20(2):144–156, 1982.

[6] W.K. Grassmann. Finding transient solutions in Markovian event systems through randomization. In W. J. Stewart, editor, *Numerical Solution of Markov Chains*, pages 357–371. Marcel Dekker, Inc., 1991.

[7] W.K. Grassmann, M.I. Taksar, and D.P. Heyman. Regenerative analysis and steady state distributions for Markov chains. *Operations Research*, 33(5):1107–1116, 85.

[8] A. Jensen. Markoff chains as an aid in the study of Markoff processes. *Skandinavsk Aktuarietidskrift*, 36:87–91, 1953.

[9] P.M. Morse. *Queues Inventories, and Maintenance: The Analysis of Operation Systems with Variable Supply and Demand.* Wiley, 1958.

[10] S. M. Ross. Approximating Transition Probabilities and Mean Occupation Times in Continous-time Markov Chains. *Probability in the Engineering and Informational Sciences*, pages 251–264, 1987.

[11] W.J. Stewart. *Introduction to the Numerical Solution of Markov Chains.* Princeton University Press, 1994.

[12] B. S. Yoon and J. G. Shanthikumar. Bounds and Approximations for the Transient Behavior of Continuos Time Markov Chains. *Probability in the Engineering and Informational Sciences*, pages 175–198, 1989.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399