Censoring Markov Chains and Stochastic Bounds

J-M. Fourneau[1,2], N. Pekergin[2,3,4] and S. Younès[2]

[1] INRIA Project MESCAL, Montbonnot, France
[2] PRiSM, University Versailles-Saint-Quentin, 45, Av. des Etats-Unis 78000 France
[3] Marin Mersenne Laboratory, University Paris 1, 90, Rue de Tolbiac, 75013 France
[4] LACL, 61 avenue Général de Gaulle 94010, Créteil, France
email : {jmf,nih,sayo}@prism.uvsq.fr

Abstract. We show how to combine censoring technique for Markov chain and strong stochastic comparison to obtain bounds on rewards and the first passage time. We present the main ideas of the method, the algorithms and their proofs. We obtain a substantial reduction of the state space due to the censoring technique. We also present some numerical results to illustrate the effectiveness of the method.

1 Introduction

Modeling systems with huge or infinite Markov chain is still a hard problem when the chain does not exhibit some regularity or symmetry which allow analytical techniques or lumping. An alternative approach is to compute bounds on the rewards we need to check against requirement. For instance we may obtain an upper bound on the loss probability and verify that this bound is smaller than the quality of service required by a network application. To compute bounds on rewards the usual way is to bound the steady-state or transient distribution at time t, define the elementary reward for state i and perform the summation of the product of the elementary rewards by the state probabilities. The last two parts are the easiest step of the method. The main difficulty is to obtain a bound of the steady state or transient distributions. Some rewards are also related to the first passage time or the absorbing time if the chain has some absorbing states. We must in that case compute the fundamental matrix of the chain, again a difficult problem when the state space is extremely large. The main idea is to derive a smaller chain which provides a bound. In the recent years, several algorithms have been published to obtain some stochastic bounds on Markov chains [18, 7, 10, 3]. But most of these algorithms have used the lumpability approach to reduce the size of chain and only considered finite DTMC (Discrete Time Markov Chain). Here we show how we can compute stochastic bounds using the Censored Markov chain and how we can deal with large Markov chains.

Consider a discrete time Markov chain $\{X_t : t = 1, 2, \ldots\}$ with finite state space S. Suppose that $S = E \cup E^c$, $E \cap E^c = \emptyset$. Suppose that the successive visits of X_t to E take place at time epochs $0 < t_1 < t_2 < \ldots <$. Then the chain $\{X^E_u = X_{t_u}, u = 1, 2, \ldots\}$ is called the censored process (or chain) with censoring set E [19]. Let Q denote the transition matrix of chain X_t. Consider the partition of the state space to obtain a block description of Q:

$$Q = \begin{array}{cc} \begin{pmatrix} Q_E & Q_{EE^c} \\ Q_{E^cE} & Q_{E^c} \end{pmatrix} & \begin{array}{c} E \\ E^c \end{array} \end{array} \tag{1}$$

The censored chain only watches the chain when it is in E. Under some structural condition on the matrix, it can be proved [19] that the stochastic matrix of the censored chain is:

$$S_E = Q_E + Q_{EE^c} \left(\sum_{i=0}^{\infty} (Q_{E^c})^i \right) Q_{E^cE} \tag{2}$$

Assume that (Q_{E^c}) does not contain any recurrent class, the fundamental matrix is $\sum_{i=0}^{\infty}(Q_{E^c})^i = (I - Q_{E^c})^{-1}$. Censored Markov chains have also been called restricted or watched Markov chains. When the chain is ergodic there are strong relations with the theory of stochastic complement [11]. Note that it is not necessary for censored Markov chains to be ergodic and we can study for instance the absorbing time. In many problems Q can be large and therefore it is difficult to compute $(I - Q_{E^c})^{-1}$ to finally get S_E. Deriving bounds of S_E from Q_E and some information on the other blocks without computing S_E is therefore an interesting alternative approach.

To the best of our knowledge this paper is the first approach to combine stochastic bounds and censored Markov chain, even if the stochastic complement approach was mentioned in a survey on algorithmic aspects of stochastic bounds [9]. However some of the methods already published for NCD (Nearly Completely Decomposable) chains may be applied to construct bounds. For instance in [17] Truffet has proposed a two-level algorithm for NCD chains by using aggregation and the stochastic ordering to compute bounding distributions. This method is different from the bounded aggregation method proposed by Courtois-Semal which uses polyhedra theory [4] to compute bounds. In [13], this approach has been extended by employing reordering to improve the accuracy and a better component-wise probability bounding algorithm. In these works, before employing the aggregation of blocks, the slack probabilities which are small due to the NCD structure are included in the last column for the upper bounding case and to the first column for the lower bounding case. In the case of general Markov chains, an algebraic approach has been recently proposed to dispatch slack probabilities [6].

In this work, we derive bounds on S_E, in a completely different way by applying graph algorithms. Indeed we propose to compute element-wise lower bounds on the second term of Eq. 2 by exploring some paths that return to partition E passing through partition E^c. We give some relations between element-wise lower bound on S_E and the derived stochastic bounds on it.

The following of the paper is as follows. In section 2 we present stochastic bounds and the basic algorithm to build a monotone upper bound for any stochastic matrix and we present the basic operator used to formally define this algorithm. We also give some necessary definitions and results for censored Markov chains. Section 3 is devoted to the main theoretical results of this paper: we show how we can obtain a stochastic upper bound of S_E from any element-wise lower bound of S_E. We also prove that the more accurate is the element-wise lower bound the more accurate is the stochastic upper bound. Clearly, Q_E is an element-wise lower bound of S_E. $Q_{EE^c}\left(\sum_{i=0}^{\infty}(Q_{E^c})^i\right)Q_{E^cE}$ represents all the paths entering E^c, staying in this set for an arbitrary number of transitions and finally returning to E. Thus if we only keep some paths in consideration we obtain again an element-wise lower bound of S_E. We develop this approach in section 4 using several graph techniques to obtain sets of paths and their probabilities. Finally in section 5 we present some examples and numerical results.

2 Theoretical background

In this section, we present some preliminaries on the stochastic comparison method and on censored Markov chains. We refer to the books [14, 15] for the theoretical issues for comparison of random variables and Markov chains. We study Discrete Time Markov chains (DTMC in the following) on finite or denumerable state space endowed with a total ordering. Let $\mathcal{S}$ be the state space.

2.1 Basic algorithms to bound a Markov chain

Definition 1. *Let X and Y be random variables taking values on a totally ordered space S. Then X is said to be less than Y in the strong stochastic sense, ($X \preceq_{st} Y$) if and only if $E[f(X)] \leq E[f(Y)]$ for all non decreasing functions $f : S \rightarrow R$, whenever the expectations exist.*

Indeed $\preceq_{st}$ ordering provides the comparison of the underlying probability distribution functions: $X \preceq_{st} Y \leftrightarrow Prob(X > a) \leq Prob(Y > a) \quad \forall a \in S$. Thus it is more probable for Y to take larger values than for X. Since the $\preceq_{st}$ ordering yields the comparison of sample-paths, it is also known as sample-path ordering. We give in the next proposition the $\preceq_{st}$ comparison in the case of finite state space.

Property 1. Let X, Y be random variables taking values on $\{1, 2, \cdots, n\}$ and p, q be probability vectors which are respectively denoting distributions of X and Y, $X \preceq_{st} Y$ iff $\sum_{j=i}^{n} p[j] \leq \sum_{j=i}^{n} q[j] \quad \forall i = \{n, n-1, \cdots, 1\}$. Remark that $X = Y$ implies that $X \preceq_{st} Y$.

The stochastic comparison of random variables has been extended to the comparison of Markov chains. It is shown in Theorem 5.2.11 of [14, p.186] that monotonicity and comparability of the probability transition matrices of time-homogeneous Markov chains yield sufficient conditions to compare stochastically the underlying chains. We first define the monotonicity and comparability of stochastic matrices and then state this theorem and some useful corollaries.

Definition 2. *Let P be a stochastic matrix. P is said to be stochastically st-monotone (monotone for short) if for any probability vectors p and q,*

$$p \preceq_{st} q \implies p\ P \preceq_{st} q\ P.$$

Definition 3. *Let P and Q be two stochastic matrices. Q is said to be an upper bounding matrix of P in the sense of the strong stochastic order ($P \preceq_{st} Q$) iff*

$$P_{i,*} \preceq_{st} Q_{i,*}, \quad \forall i$$

where $P_{i,*}$ denotes the i^{th} row of matrix P.

Theorem 1. *Let P (resp. Q) be the probability transition matrix of the time-homogeneous Markov chain $\{X_t, t \geq 0\}$ (resp. $\{Y_t, t \geq 0\}$). If*

- *$X_0 \preceq_{st} Y_0$,*
- *at least one of the probability transition matrices is monotone, that is, either P or Q is monotone,*
- *the transition matrices are comparable, (i.e. $P \ \preceq_{st}\ Q$).*

then $X_t \preceq_{st} Y_t \quad \forall t$.

Then the following corollary ([14]) lets us compare the steady-state distributions of Markov chains when they exist. And we can also compare absorption time if the chain has an absorbing state (see [2] for a proof).

Corollary 1. *Let Q be a monotone, upper bounding matrix for P for the st-ordering. If the steady-state distributions ($\mathbf{\Pi_P}$ and $\mathbf{\Pi_Q}$) exist, then $\mathbf{\Pi_P} \preceq_{st} \mathbf{\Pi_Q}$.*

Corollary 2 (proposition 2.9 in [2])). *Let $\{X_t, t \geq 0\}$ and $\{Y_t, t \geq 0\}$ be two Markov chains on the same finite state space. Assume that the last state (say n) is absorbing for both chains. Assume that $X_t \preceq_{st} Y_t$, $\forall t$ then $T_{i,n}(Y) \preceq_{st} T_{i,n}(X)$ where $T_{i,n}(X)$ is the absorption time in n for chain X when initial state is i.*

Stochastic comparison and monotonicity can be represented by linear inequalities. Once we have derived a set of equalities, instead of inequalities and ordered them we obtain a constructive way to design a monotone upper bounding stochastic matrix Q for an arbitrary stochastic matrix P.

$$\begin{cases} \sum_{k=j}^{n} Q_{1,k} &= \sum_{k=j}^{n} P_{1,k} \\ \sum_{k=j}^{n} Q_{i+1,k} &= max(\sum_{k=j}^{n} Q_{i,k}, \sum_{k=j}^{n} P_{i+1,k}) \quad \forall\ i,j \end{cases} \tag{3}$$

This is the basic theory behind Vincent's algorithm [1]. We now present an operator description of this basic algorithm (see [9] for a survey of algorithmic aspects of stochastic bounds).

Definition 4. *Let P and Q be two positive matrices with the same size, $P \preceq_{el} Q$ iff $P[i,j] \leq Q[i,j]$ for all i and j.*

Definition 5. *Following the presentation in [5] we define two operators* r *and* v *for matrix of size* $n \times n$ *as follows:*

- *r is a summation operator:* $r(P)[i,j] = \sum_{k=j}^{n} P[i,k]$. *The inverse of* r *(denoted as* r^{-1}*) is:*

$$r^{-1}(P)[i,j] = \begin{cases} P[i,n] & \text{if } j = n \\ P[i,j] - P[i,j+1] & \text{if } j < n \end{cases}$$

- *Let* v *be the operator defined by:*

$$v(P)[i,j] = max_{m \leq i}(\sum_{k \geq j} P[m,k]) = max_{m \leq i} r(P)[m,j] \tag{4}$$

Property 2. Vincent's algorithm is simply operator $r^{-1}v$.

Example 1. Let P be a stochastic matrix. Vincent's algorithm gives:

$$P = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.4 & 0.2 & 0.3 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.2 & 0 & 0.4 & 0.4 \end{bmatrix} \quad v(P) = \begin{bmatrix} 1 & 0.9 & 0.6 & 0.4 \\ 1 & 0.9 & 0.6 & 0.4 \\ 1 & 0.9 & 0.7 & 0.4 \\ 1 & 0.9 & 0.8 & 0.4 \end{bmatrix} r^{-1}v(P) = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.2 & 0.3 & 0.4 \\ 0.1 & 0.1 & 0.4 & 0.4 \end{bmatrix}$$

Property 3. Let P and Q two stochastic matrices with the same size, $P \preceq_{st} Q$ iff $r(P) \preceq_{el} r(Q)$.

And we define two new operators θ and γ which transform a sub-stochastic matrix P into stochastic matrix by adding the probability missing in P in the last (resp. the first) column.

$$\theta(P)[i,j] = \begin{cases} P[i,j] & if\ j < n \\ P[i,j] + \beta_i & if\ j = n \end{cases} \quad \gamma(P)[i,j] = \begin{cases} P[i,j] & if\ j > 1 \\ P[i,j] + \beta_i & if\ j = 1 \end{cases}$$

where $\beta_i = 1 - \sum_{j=1}^{n} P[i,j]$. Of course, if P is stochastic $\theta(P) = P = \gamma(P)$.

2.2 Censoring a Markov chain

Let us go back to the definition and the fundamental results on censored chains.

Lemma 1 (Theorem 2 in [19]). *Let* Q *be the transition probability matrix of a DTMC* $X(t)$*. Consider a partition of the finite state space* $\mathcal{S}$ *into two subsets* E *and* E^c*.*

$$Q = \begin{array}{cc} \begin{pmatrix} Q_E & Q_{EE^c} \\ Q_{E^cE} & Q_{E^c} \end{pmatrix} & \begin{array}{c} E \\ E^c \end{array} \end{array}$$

Then, the censored process $X_E(t)$ is a Markov chain and its transition probability matrix is given by:

$$S_E = Q_E + Q_{EE^c}\left(\sum_{i=0}^{\infty}(Q_{E^c})^i\right)Q_{E^cE} \tag{5}$$

When Q is irreducible the transition probability matrix of the censored chains is the stochastic complement matrix defined by Meyer and we have the following results [11]:

Theorem 2. *If Q is irreducible, with steady state distribution $\mathbf{\Pi}_Q = (\mathbf{\Pi}_E, \mathbf{\Pi}_{E^c})$ and transient distribution at discrete time epoch t, $\mathbf{\Pi}_Q^t = (\mathbf{\Pi}_E^t, \mathbf{\Pi}_{E^c}^t)$. Then the steady state distribution $\mathbf{\Pi}_{S_E}$ and the transient distribution at time t, $\mathbf{\Pi}_E^t$ of the censored matrix S_E are given by:*

$$\mathbf{\Pi}_{S_E} = \frac{\mathbf{\Pi}_E}{\sum_{i\in E}\mathbf{\Pi}_E(i)} \quad \textit{and} \quad \mathbf{\Pi}_{S_E}^t = \frac{\mathbf{\Pi}_E^t}{\sum_{i\in E}\mathbf{\Pi}_E^t(i)} \tag{6}$$

The transition matrix of the censored chain can be decomposed into two parts. Q_E is an element-wise lower bound of S_E. $Q_{EE^c}\left(\sum_{i=0}^{\infty}(Q_{E^c})^i\right)Q_{E^cE}$ represents all the paths entering E^c, staying in this set for an arbitrary number of transitions and finally returning to E. Assuming that Q_{E^c} does not contain any recurrent class we have: $\sum_{i=0}^{\infty}(Q_{E^c})^i = (Id - Q_{E^c})^{-1}$. But the state space is so huge that this operation is too complex. Thus instead of computing S_E we advocate that we can obtain stochastic bounds of this matrix.

As $Q_{EE^c}\left(\sum_{i=0}^{\infty}(Q_{E^c})^i\right)Q_{E^cE}$ contains all the paths, we will only keep some of them in consideration and we obtain more accurate bounds of S_E. The main idea is that only some elements of Q_{E^c} are generated and stored during the construction of the Markov chain.

3 Bounds for Censored chains

We first prove some technical lemmas and then give the theorems to provide bounds.

Lemma 2. *Let P and Q be two stochastic or sub-stochastic matrices of size $n \times n$. if $P \preceq_{el} Q$ then $\theta(Q) \preceq_{st} \theta(P)$.*

Proof. $\forall \quad 1 \leq i \leq n$ and $\forall \quad 1 \leq q \leq n$ we have:

$$\sum_{j=q}^{n}\theta(Q)[i,j] = 1 - \sum_{j<q}Q[i,j] \leq 1 - \sum_{j<q}P[i,j] = \sum_{j=q}^{n}\theta(P)[i,j]$$

Thus $\theta(Q) \preceq_{st} \theta(P)$.

Lemma 3. *Let P and Q be two stochastic matrices, if $P \preceq_{st} Q$ then $r^{-1}v(P) \preceq_{st} r^{-1}v(Q)$.*

Proof. It follows from Property 3 and Eq. 4 that if $P \preceq_{st} Q$ then $v(P) \leq_{el} v(Q)$. This implies following Property 3 that $r^{-1}v(P) \preceq_{st} r^{-1}v(Q)$.

We now present the two fundamental theorems which allow to bound a censored Markov chain. For both theorems, let $\{X_t, t \geq 0\}$ be a denumerable DTMC with transition matrix Q and E a finite subset of state space S. Let S_E be the matrix of the censored Markov chain and Q_E the block of Q restricted to states in E.

Theorem 3. *For all sub-stochastic matrix M such that $Q_E \preceq_{el} M \preceq_{el} S_E$, we have*

$$S_E \preceq_{st} r^{-1}v\theta(M) \tag{7}$$

Proof. We assume that $M \preceq_{el} S_E$. We apply Lemma 2 to obtain: $\theta(S_E) \preceq_{st} \theta(M)$. But S_E is a stochastic matrix. Thus $\theta(S_E) = S_E$ and we get: $S_E \preceq_{st} \theta(M)$. We now apply lemma 3 which implies that: $r^{-1}v(S_E) \preceq_{st} r^{-1}v\theta(M)$.

We finally remark that due to the definition of operators r and v we have $S_E \preceq_{st} r^{-1}v(S_E)$ and we obtain that $S_E \preceq_{st} r^{-1}v\theta(M)$ to complete the proof.

Similarly we can obtain a lower bound with operator γ and the modified version of Vincent's algorithm to obtain monotone lower bound. The following theorem explains how we can improve this bound. If we are able to improve the element-wise lower bound M, we also improve the stochastic upper bound for S_E. However remember that $X \preceq_{st} Y$ does not exclude that $X = Y$ and the improvement on the stochastic bound can be zero (see for instance the first part of the example in Section 4).

Theorem 4. *For all sub-stochastic matrices $M1$ and $M2$ such that $M1 \preceq_{el} M2 \preceq_{el} S_E$, we have:*

$$r^{-1}v\theta(M2) \preceq_{st} r^{-1}v\theta(M1) \tag{8}$$

Proof. As we assume that $M1 \preceq_{el} M2$ Lemma 2 shows that: $\theta(M2) \preceq_{st} \theta(M1)$. But $\theta(M1)$ and $\theta(M2)$ are stochastic matrices. We then apply Lemma 3 to obtain $r^{-1}v\theta(M2) \preceq_{st} r^{-1}v\theta(M1)$ to complete the proof.

So the algorithms mainly consist in computing an element-wise lower bound of S_E which is obtained by adding some probability to Q_E and then apply operators θ and Vincent's algorithm. We show in the next section how we can improve element-wise lower bounds of S_E. We now show that bounds on censored chains can provide bounds for some performances measures on the original chain. First we have a very simple property, the proof of which is a simple application of theorem 2.

Property 4. Let i and j be two states in S. If $i \in E$ and $j \in E^c$ then:

$$\mathbf{\Pi}_Q(i) \leq \mathbf{\Pi}_{S_E}(i) \;\; \textit{and} \;\; \mathbf{\Pi}_Q(j) \leq \mathbf{\Pi}_{S_{E^c}}(j) \tag{9}$$

We can derive bounds for steady-state rewards, absorbing probabilities and absorbing time. Assume that we have derived two monotone stochastic matrices UE and LE such that: $LE \preceq_{st} S_E \preceq_{st} UE$.

Property 5 (Steady-state rewards). Let $w : S \to R$ be the reward function that assign to each state $i \in S$ a reward value $w(i)$. Assume that $w(i) \geq 0$ for all i. Let E be the set of states which has a positive reward. Assuming that we sort the states in E such that function w is non decreasing. We clearly have:

$$R = \sum_{i \in S} w(i)\mathbf{\Pi}_Q(i) = \sum_{i \in E} w(i)\mathbf{\Pi}_E(i) \leq \sum_{i \in E} w(i)\mathbf{\Pi}_{S_E}(i) \leq \sum_{i \in E} w(i)\mathbf{\Pi}_{UE}(i)$$

We obtain an upper bound on the reward.

Property 6 (Probability of Absorption). We consider a chain with a finite number of absorbing states. Assume that all these absorbing states are in E and assume that the initial state is in E. Assume also that the states which immediately precede absorbing states are also in E. Then the absorption probabilities in the initial chain and in the censored chain are the same.

Proof. Remember that when we have a block decomposition of a transition matrix with absorbing states equal to $\left[\begin{array}{c|c} Id & 0 \\ \hline F & H \end{array}\right]$, matrix $M = (Id - H)^{-1}$ exists and is called the fundamental matrix [16]. Furthermore the entry $[i, j]$ of the product matrix $M * F$ gives the absorption probability in j knowing that the initial state is i.

We assume that the absorbing states are gathered in the first part of set E. Thus we can describe the matrix of the chain by its block decomposition:

$$\left[\begin{array}{c|c||c} Id & 0 & 0 \\ \hline R & A & B \\ \hline\hline 0 & C & D \end{array}\right]$$

According to lemma 1, the transition matrix of the censored chain is:

$$\left[\begin{array}{c|c} Id & 0 \\ \hline R & A \end{array}\right] + \left[\begin{array}{c} 0 \\ \hline B \end{array}\right] \sum_i [D]^i \left[\begin{array}{c|c} 0 & C \end{array}\right]$$

which is finally equal to: $\left[\begin{array}{c|c} Id & 0 \\ \hline R & A + B\sum_i D^i C \end{array}\right]$. As D is transient, we have: $\sum_i D^i = (Id - D)^{-1}$. And the fundamental matrix of the censored chain is:

$$(Id - A - B(Id - D)^{-1}C)^{-1}.$$

The fundamental matrix of the initial chain is: $M = \left[\begin{array}{c|c} Id - A & B \\ \hline C & Id - D \end{array}\right]^{-1}$. To obtain the probability we must multiply by $\left[\begin{array}{c} R \\ \hline 0 \end{array}\right]$ and consider an initial state in E. Thus we only have to compute the upper-left block of F. According to [12] page 123, it is equal to:

$$(Id - A - B(Id - D)^{-1}C)^{-1}$$

if blocks $(Id - A)$ and its Schur complements are non singular. This is clearly true. So we have the same absorption probability in Q and in S_E and bounds for the censored chain will also be bounds for the initial chain.

Property 7 (Average Time for Absorption). We consider a chain X with several absorbing states and the same block decomposition. Let Y be the censored chain. Let i be an initial state in E, j an arbitrary state in $\mathcal{S}$ and k an absorbing state. Let $Z_X[i,j]$ be the average number of passages in j before absorption knowing that the initial state is i for chain X. We have:

1. $Z_X[i,j] = Z_Y[i,j]$ if j is in E.
2. $\mathbf{E}(T_{i,k}(Y)) \leq \mathbf{E}(T_{i,k}(X))$.

Proof. Again remember that the average number of visits in j when the initial state is i is entry $[i,j]$ of the fundamental matrix. The proof of the previous property states that the upper-left block of the fundamental matrix of X is equal to the fundamental matrix of Y. This equality implies the first part of the property. The second part is a consequence of the first part and of the average number of visits to states in E^c which are positive in X and equals to 0 in Y.

4 Algorithms

The algorithms must find some paths which are contained in the fundamental matrix $(Id - Q_{E^c})^{-1}$, thus there is clearly a trade-off between complexity and accuracy. So we have developed several algorithms and data structures to deal with paths exploration. The aim is to deal with chains which are so large that the transition matrix does not fit in memory.

The algorithms compute some paths leaving immediately state i in E and coming back to E in any state. The output of the algorithms is a row vector called q whose the jth entry contains the probability of the paths from i to j which have been selected. Thus if we add q to row i of Q_E we obtain a more accurate element-wise lower bound for row i of S_E : $Q_E[i,*] \leq_{el} Q_E[i,*] + q \leq_{el} S_E[i,*]$.

Note also that all the rows do not have the same importance for the computation of the bound. Due to the monotonicity constraints, the last rows are often completely modified by Vincent's algorithm. Thus it is much more efficient to try to improve the first rows of Q_E than the last ones. This is illustrated by the following example.

$$Q_E = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.1 \\ 0.1 & 0.4 & 0.2 & 0 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.3 & 0 & 0.4 & 0 \end{bmatrix}$$

Truffet's approach gives:

$$\theta(Q_E) = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.4 & 0.2 & 0.3 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.3 & 0 & 0.4 & 0.3 \end{bmatrix} \qquad r^{-1}v\theta(Q_E) = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.2 & 0.3 & 0.4 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{bmatrix}$$

Now suppose that one has computed the probability $[0.1, 0.1, 0, 0.1]$ of some paths leaving E from state 4 and entering again set E after a visit in E^c. This is a lower bound of the set of all paths beginning in state 4. Let M be the improved element-wise lower bound.

$$M = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.1 \\ 0.1 & 0.4 & 0.2 & 0 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.4 & 0.1 & 0.4 & 0.1 \end{bmatrix} \quad \theta(M) = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.4 & 0.2 & 0.3 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.4 & 0.1 & 0.4 & 0.1 \end{bmatrix} \quad r^{-1}v\theta(M) = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.3 & 0.2 & 0.4 \\ 0.1 & 0.2 & 0.3 & 0.4 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{bmatrix}$$

And the bound does not change despite the computation of paths beginning in state 4. Assume now one has improved the first row and we have got the same vector of probability for the paths: $[0.1, 0.1, 0, 0.1]$.

$$M = \begin{bmatrix} 0.2 & 0.4 & 0.2 & 0.2 \\ 0.1 & 0.4 & 0.2 & 0 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.3 & 0 & 0.4 & 0 \end{bmatrix} \quad \theta(M) = \begin{bmatrix} 0.2 & 0.4 & 0.2 & 0.2 \\ 0.1 & 0.4 & 0.2 & 0.3 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.3 & 0 & 0.4 & 0.3 \end{bmatrix} \quad r^{-1}v\theta(M) = \begin{bmatrix} 0.2 & 0.4 & 0.2 & 0.2 \\ 0.1 & 0.4 & 0.2 & 0.3 \\ 0.1 & 0.2 & 0.4 & 0.3 \\ 0.1 & 0.2 & 0.4 & 0.3 \end{bmatrix}$$

Clearly this bound is now much better than the original one.

We consider the directed graph $G = (\mathcal{S}, DE)$ associated to the initial Markov chain where DE is the set of directed edges. If $Q(i, j) > 0$ then there exists an arc from i to j in G and arc (i, j) has probability $Q(i, j)$. The directed edges (or arcs) in the graph are labelled with a positive cost. A path $\mathcal{P}$ is an ordered list of consecutive arcs. The cost of path $\mathcal{P}$ is the sum of the cost of arcs which belong to $\mathcal{P}$ multiplied by their number of occurrences in $\mathcal{P}$. Indeed an arc may appear several times in a path. The probability of path $\mathcal{P}$ (denoted as $Pr(\mathcal{P})$) is the product of the probability of the arcs which belongs to $\mathcal{P}$. Again we must take into account the number of occurrence of the arcs in the path. The set of paths beginning in i a node of E, then jumping to a node in E^c, staying in E^c for an arbitrary number of jumps and finally entering again E in state j is denoted as $\mathcal{SP}_{i,j}$.

The main idea is to select paths with high probabilities and to perform this selection very quickly. We present here two type of algorithms: the Shortest Path approach and Breadth First search. The first one builds one path to every destination while the second builds all the paths whose lengths are smaller than a parameter Δ. We also show that we can take into account the self loops to obtain easily an infinite set of paths rather than a single one.

4.1 Shortest Path

We use Dijkstra's shortest path algorithm. The length taken into account in the algorithm is the cost $c()$ which is positive. The following property states how we first compute the cost to obtain the path with the highest probability.

Property 8. If for all arcs in DE, the cost is defined as $c(i, j) = -log(Q(i, j))$ then the shortest path according to cost c is also the path with maximum probability. Note that as $Q(i, j) < 1$ the cost is positive.

Proof. Let $\mathcal{P}1$ be this shortest path. Assume that there exists $\mathcal{P}2$, a path such that $Pr(\mathcal{P}2) > Pr(\mathcal{P}1)$. As function logarithm is increasing we have: $log(Pr(\mathcal{P}2)) > log(Pr(\mathcal{P}1))$. The probability of the path is the product of the probability of arcs. Thus:

$$\sum_{(i,j)\in\mathcal{P}2} log(Q(i,j)) > \sum_{(k,l)\in\mathcal{P}1} log(Q(k,l))$$

After substitution:

$$\sum_{(i,j)\in\mathcal{P}2} c(i,j) < \sum_{(k,l)\in\mathcal{P}1} c(k,l)$$

And $\mathcal{P}2$ is shorter than the shortest path; a contradiction.

Thus the algorithm searches the shortest path from state i in a graph where the costs are defined as the negative of the logarithm of transition probabilities and where the arcs from i to other nodes in E have been removed because we want to get a path from i which passes through E^c and comes back to E. The shortest path with the cost function may have a large number of arcs. Thus we must give a bound on the number of arcs in the shortest path to avoid very large number of iterations. Let Δ be this bound. The following algorithm computes the probability of the shortest path. In the algorithm, $\boldsymbol{P}$ is the set of generated vertices, $\Gamma(x)$ denotes the set of successors of node x and $p(x)$ is the probability to reach x from i in the selected paths.

```
Algorithm 1: Shortest path
  Input  : vertex $i \in E$; $\Delta$
  Output : row vector $q$: $q_j$ is the probability to return to $j \in E$ from $i$
  $\boldsymbol{P} = \varnothing$; $q_z = 0$ , $z \in E$
  foreach vertex $x \in \Gamma(i)$ such that $x \in E^c$ do
      p(x)=prob. of transition from $i$ to $x$; put $x$ in $\boldsymbol{P}$
  end
  repeat
      Select a leaf $y \in \boldsymbol{P}$ such that $p(y) = \max_{\text{leaf } x\in\boldsymbol{P}}\{p(x)\}$
      foreach $z \in \Gamma(y)\backslash\{y\}$ do
          $p_2$ =(prob. of transition from $y$ to $z$) $* p(y)$
          switch $z$ do
              case $z \in E$: $q_z = q_z + p_2$
              case $z \notin \boldsymbol{P}$: $p(z) = p_2$; put $z$ in $\boldsymbol{P}$
              case $z$ is a leaf and $p(z) < p_2$: $p(z) = p_2$
          end
      end
  until number of iteration $> \Delta$;
```

Note that as we only search for successors of a limited number of nodes, only a part of the transition matrix must be in memory. Even if the whole matrix does not fit in memory it is sufficient that the states we really use during the construction of the paths can be stored or generated.

4.2 Adding Self Loops

Once a path from i to j is selected it is possible to build an infinite set of paths from i to j and to sum their probabilities in a closed-form formula. We just have to use the directed cycles. The proofs of following properties are quite simple and they are omitted here due to the limitation on the size of the paper.

Property 9. Let i and j be two arbitrary vertices in E. Let $\mathcal{P}$ be a path in $\mathcal{SP}_{i,j}$ with a probability p. Let k be a vertex which belongs to $\mathcal{P}$ and E^c such that there exists a directed cycle using nodes of E^c going through k. Let q be the product of the probabilities of the arcs in this directed cycle. Then the path $\mathcal{P}_k$ built with $\mathcal{P}$ and k times the directed cycle is also in $\mathcal{SP}_{i,j}$ and its probability is $p\ q^k$. Considering all these paths $\mathcal{P}_k$ for all values of k, we finally obtain a probability equals to $\frac{p}{1-q}$.

Computing a directed cycle may be difficult but it is quite simple to take into account the self loops during the visits. Indeed self loops are directed cycles and finding them does not require any new computational effort.

Property 10. Let $\mathcal{P} = (i, k_1, k_2, \ldots, k_l, j)$ be an arbitrary elementary path in $\mathcal{SP}_{i,j}$. Suppose that every vertex k_m in the path has a self loop with probability q_{k_m}. If there is no loop in k_m we simply have $q_{k_m} = 0$. Then all the path obtained from $\mathcal{P}$ and an arbitrary number of visits in each loop is also in $\mathcal{SP}_{i,j}$. And the resulting probability for all these paths is $p \prod_{m=0}^{l} \frac{1}{1-q_m}$.

4.3 Breadth First Search

We just build all paths of length smaller than Δ using a Breadth First search technique and computing their probabilities. Some of these paths return to a node in E at step $k \leq \Delta$. We use in the algorithm the same notation for data structure as in Shortest Path Algorithm and we finally denote by $InE[y,z]$ the probability to enter E through z leaving from $y \in E^c$.

5 Examples and Numerical Results

Due to the limitation on the size of the paper, it is not possible to present here a real example. We have just designed an abstract model to test our algorithms and show some numerical experiments. We consider a set of N resources: they can be operational or faulty. In the considered model we distinguish two types of faults: hard and soft, that we denote respectively by h and s. The fault arrivals of (h and s) follow independent Poisson processes with rate respectively λ_h and λ_s. The distribution of times to fix a fault are exponential with rate μ_h and μ_s except when all the resources are faulty. In that case, the repairman can speed up

Algorithm 2: Breadth-First search

```
Input    : vertex $i \in E$; $\Delta$
Output : row vector $q$ such that $q_z$ is the probability to return to $z \in E$
$\boldsymbol{P} = \varnothing$;
foreach $x \in \Gamma(i)$ such that $x \in E^c$ do
    p(x)=prob. of transition $i$ to $x$; put $x$ in $\boldsymbol{P}$
end
$\boldsymbol{P}_{last} = \boldsymbol{P}$;
repeat
    $\boldsymbol{P}_2^{last} = \varnothing$
    foreach vertex $x \in \boldsymbol{P}_{last}$ do
        foreach vertex $y \in \Gamma(x)$ do
            switch $y$ do
                case $y \in E$: $InE[x, y]$ =prob. of transition from $x$ to $y$
                case $y \notin \boldsymbol{P}$: $p(y) = p(x)*$(prob. of transition from $x$ to $y$); put $y$
                in $\boldsymbol{P}$ and in $\boldsymbol{P}_2^{last}$
                case $y \in \boldsymbol{P}$:  $p(y) = p(y) + p(x)*$(prob. of transition from $x$ to
                $y$); put $y$ in $\boldsymbol{P}_2^{last}$
            end
        end
    end
    $\boldsymbol{P}_{last} = \boldsymbol{P}_2^{last}$
until number of iteration $> \Delta$;
foreach  $y \in \boldsymbol{P}$ do  foreach $z \in E$ do $q_z = p(y) * InE[y, z]$
```

the fixing and with rate μ all the resources are repaired. Under these Markovian arrival hypothesis, the considered system can be modelled as a CTMC with state space $S = \{(n_s, n_h), C = n_s + n_h \leq N\}$ where C represents the total number of faulty resources, n_s (resp. n_h) represents the number of faulty resources caused by soft (resp. hard) error. The size of the underlying chain is $\frac{(N+1)(N+2)}{2}$. Note that the considered chain is not NCD because of the numerical values of rate μ we have considered in the examples.

We present in Table 1, the conditional probability p to have the N resources operational and the upper bound on this probability. The censored state space contains states with no faulty hardware components. The states are ordered according to the decreasing number of software faulty components. The second step is to determine an element-wise lower bound to S_E. We apply the shortest path algorithm presented previously in subsection 4.1 with considering self loops. Remind that we have to fix the maximum number of arcs of shortest paths Δ and the number of first rows R in which we will apply the algorithm to simplify the computation of the bound. In the following table we present results for different values of Δ and R that represent parameters of the algorithm given in column *algorithm parameters*. Numerical Results are computed in a 3.2 GHz Intel Pentium 4 CPU with 1.5 Go of memory under Linux 2.6.8 kernel system. We also report computation time T (in *second*) needed to obtain the exact and bounding probability. We can see obviously that computation times

model size		Exact		algorithm parameters		Bound	
N	space size	p	T	Δ	R	p	T
100	5151	3.622e-6	1.57	2	$N/4$	4.14361e-6	.06
					N	4.14351e-6	.08
				10	$N/4$	4.12148e-6	.07
					N	4.12111e-6	.17
300	45451	1.224e-6	32.56	2	$N/4$	1.408871e-6	.16
					N	1.40884e-6	.22
				10	$N/4$	1.40141e-6	.23
					N	1.40127e-6	.51
500	125751	7.528e-7	168.47	2	$N/4$	8.76306e-7	.27
					N	8.76287e-7	.39
				10	$N/4$	8.71677e-7	.38
					N	8.71588e-7	.91
1000	501501	4.013e-7	603.14	2	$N/4$	4.82768e-7	.78
					N	4.82757e-7	1.01
				10	$N/4$	4.80213e-7	1.08
					N	4.80164e-7	2.31
10000	50015001	-	-	2	$N/4$	2.90822e-7	55.98
					N	2.90815e-7	55.10
				10	$N/4$	2.89280e-7	71.02
					N	2.89250e-7	123.01

Table 1. $\lambda_s = 0.5$, $\lambda_h = 0.0001$, $\mu_s = \mu = 1$, $\mu_h = 0.02$

are drastically reduced using the proposed bounding approach. It also provides results when the exact analysis fails ($N = 10000$). Moreover, obtained results confirm that it is not necessary to apply proposed algorithms to all rows. For this example we do not remark a notable difference between bounds obtained by considering all rows $R = N$ or ($R = N/4$). We can therefore decrease the complexity of the computation of the bounds by considering only some rows.

6 Concluding Remarks

We have proposed a new method to numerically obtain simple stochastic bounds. This method may also help to find lower bound on the absorption time if the chain is absorbing. The chain may be very large. We are still working to improve this approach to infinite DTMC. Indeed, we must correctly define ordering and censoring for transient and ergodic infinite DTMC. We only require that set E must be finite and that the absorbing states must be observed. The method only samples some paths in the non-observed part of the chain. This allows several tradeoffs between accuracy and computation time. We proved that if we add a new path in the samples the new bound we obtain is stochastically smaller than the previous one when we compute upper bound. We hope that this new approach will open new perspectives to study very large Markov chains.

References

1. Abu-Amsha, O., Vincent, J.M.: An algorithm to bound functionals of Markov chains with large state space. In: *4th INFORMS Conference on Telecommunications*, Boca Raton, Florida, (1998)
2. M. Benmammoun, A. Busic, J.M. Fourneau and N. Pekergin: Increasing convex monotone Markov chains: theory, algorithms and applications. *Markov Anniversary Meeting*. Boson Books, pp 189-210, 2006.
3. A. Busic, J.M. Fourneau. Bounds for Point and Steady-State Availability: An Algorithmic Approach Based on Lumpability and Stochastic Ordering. *EPEW 05*, LNCS 3670, Versailles, 2005 , pp 94-108.
4. P. Courtois, P. Semal: Bounds for the positive eigenvectors of nonnegative matrices and for their approximations by decomposition. *J. of ACM* Vol 31 pp. 804-825 (1984).
5. T. Dayar, J.M. Fourneau and N. Pekergin: Transforming stochastic matrices for stochastic comparison with the st-order. *RAIRO-RO*, V 37, 2003, pp 85-97.
6. T. Dayar, N. Pekergin, and S. Younes: Conditional Steady-State Bounds for a Subset of States in Markov Chains, *SMCTools*, Pisa,Italy, 2006.
7. J.M. Fourneau, M. Le Coz and F. Quessette: Algorithms for an irreducible and lumpable strong stochastic bound. *Linear Algebra and Applications*, Vol 386, pp 167-186, 2004.
8. J.M. Fourneau, M. Le Coz, N. Pekergin and F. Quessette: An open tool to compute stochastic bounds on steady-state distributions and rewards. *IEEE Mascots* 03, Orlando, USA, 2003.
9. J.M. Fourneau and N. Pekergin. An algorithmic approach to stochastic bounds. *Performance evaluation of complex systems: Techniques and Tools*. LNCS 2459, pp. 64-88, 2002.
10. S. Haddad, P. Moreaux: Sub-stochastic matrix analysis for bounds computation-Theoretical results. *Eur. Jour. of Operational. Res.* Vol.176, pp 999-1015, 2007.
11. C.D. Meyer: Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Review*. Vol.31, N2 pp. 240-272, 1989.
12. C.D. Meyer: *Matrix Analysis and Applied Linear Algebra*, SIAM, 2000.
13. N. Pekergin, T. Dayar, D. Alparslan: Compenent-wise bounds for nearly completely decomposable Markov chains using stochastic comparison and reordering. *Eur. Jour. of Op. Res.* Vol 165, pp 810-825, 2005.
14. A. Muller and D. Stoyan: *Comparison Methods for Stochastic Models and Risks*, Wiley , New York, 2002.
15. M. Shaked and J.G. Shantikumar: *Stochastic Orders and Their Applications*, Academic Press, San Diago, 1994.
16. K.S. Trivedi: *Probability and Statistic with Reliability, Queueing and Computer Science Applications*. Second Edition, Wiley, 2002.
17. L. Truffet: Near Complete Decomposability: Bounding the error by a Stochastic Comparison Method. Ad. in *App. Prob.*, Vol. 29, pp.830-855, 1997.
18. L. Truffet: Reduction Technique For Discrete Time Markov Chains on Totally Ordered State Space Using Stochastic Comparisons. *Journal of Applied Probability*, Vol. 37, N3, 2000.
19. Y.Q. Zhao and D. Liu: The Censored Markov chain and the Best Augmentation. *Jour. of App. Prob.*, Vol. 33 pp. 623-629, 1996.