Université de Reims
Champagne-Ardenne

Sub-stochastic matrix analysis and performance bounds

Serge Haddad, Patrice Moreaux

CReSTIC
UFR Sciences de Reims
B.P. 1039
51687 REIMS Cedex 02
FRANCE

Research Report
RAP-CReSTIC-1- (15-10-04)
October 2004

Résumé

D'une part, l'espace d'états des modèles markoviens complexes peut fréquemment être partitionné en un sous-ensemble de faible taille mais de grande probabilité et un grand sous-ensemble de faible probabilité. D'autre part, l'évaluation de performances et l'analyse de fiabilité impliquent le calcul d'indices de performances souvent définis comme des fonctions de récompense instantanée sur les états du modèle. Un des moyens de réduire les complexités spatiales et temporelles de ces calculs seraient consiste à éviter de représenter explicitement une partie importante du sous-ensemble de faible probabilité. Dans ce rapport, nous présentons une méthode pour déduire des bornes de telles récompenses directement à partir des paramètres du modèle - vitesses de transition ou probabilités. Cette méthode est fondée sur l'analyse d'une chaîne de Markov agrégée et sur les propriétés de la comparaison au sens de l'ordre stochastique fort pour les chaînes à temps discret et celles à temps continu. Nous proposons également une méthode spécifique lorsque la récompense est la vitesse de sortie d'un sous-ensemble d'une chaîne de Markov à temps continu. Nous illustrons notre approche par quelques exemples qui démontrent son intérêt.

Mots-clés : Bornes stochastiques, processus markovien, processus stochastique, récompense.

Abstract

On the one hand, the state space of complex Markovian models can often be partitioned between a small subset with a high steady-state probability and a large subset with a low steady-state probability. On the other hand, performance evaluation and reliability analysis require the computation of performance indices, often defined as functions of instantaneous rewards on the states of the model. Thus the time and space complexity of this computation would be greatly decreased by avoiding the explicit representation of a large part of the subset associated to the low probability. In this report, we present a method to derive bounds on such rewards directly from bounds on the parameters of the model - transition rates or probabilities. The method is based on the analysis of an aggregated Markov chain and on the properties of strong stochastic comparison for discrete as well as continuous Markov (sub-)chains. We also propose a specific method when the reward is the output rate towards a subset of states of a continuous Markov chain. Finally we illustrate our approach on some examples in order to show its interest.

Key words: Markovian process, reward, stochastic process, stochastic bounds.

Contents

List of Figures

List of Tables

1 Introduction

It is well-known that in many complex systems with (discrete or continuous) Markovian behaviour, the steady-state probability mass is concentrated on a small subset of states. Thus in order to reduce the combinatory explosion, starting from the model one would like to develop an a priori aggregated Markov chain on the irrelevant states w.r.t. the probability distribution. Ideally the steady-state probability of the reduced chain should be the aggregation of the original steady-state probability. Such a chain exists and is called the exact aggregation of the original chain. However since the rates between aggregated states involves the computation of the original steady-state probability (i.e. $q_{i,j} = \pi(S_i)^{-1} \cdot \Sigma_{s \in S_i} \pi(s) \cdot \Sigma_{s' \in S_j} q_{s,s'}$), this exact aggregation seems to be useless.

Fortunately in some typical cases, knowing bounds on these rates is sufficient to deduce bounds on the steady-state probabilities. As one of the possible applications of our work is a better handling of such cases, we briefly describe in figure 1 a generic model and the appropriate bounding algorithm. In the model, there is one state variable (e.g. the number of failed machines or the number of remote procedure calls) which induces a partition of states $S = \biguplus_{k=1}^{K} S_k$ such that the probability mass quickly decreases w.r.t. this parameter. Moreover in a state transition the variable can arbitrarily increase whereas it can only decrease by one unit (e.g. no simultaneous achievements of repair or call). Following the notations of the figure, we informally justify the method (see the references for a more detailed presentation).

1. Given a subset of states, one substitutes a Markov chain (MC) X by a family of MC X_i indexed by the entry points of the subset where in each X_i, all the entries in the subset are redirected to s_i. Then the steady-state probability of X is a barycenter of the family of the steady-state probabilities corresponding to X_i (see [3, 4]).

2. The second step is simply the application of the exact aggregation on X_i producing Y_i such that the steady-state probability vector restricted to $\biguplus_{k=1}^{k0} S_k$ in the MC X_i is identical to the corresponding vector in Y_i

3. The last step is specific to this kind of MC and states that the steady-state probability vector restricted to $\biguplus_{k=1}^{k0} S_k$ in the MC Z_i is a lower bound of the corresponding vector in Y_i if $\forall i,j\ \lambda^+_{i,j} \geq \lambda_{i,j}$ and $\mu^-_{i,j} \leq \mu_{i,j}$ (see [12, 9]).

Summarizing the method, one firstly obtains the bounds on the firing rates by a structural analysis of the model, then one computes the steady-state probabilities of the MCs Z_i and finally one deduces a lower bound on the probability vector of the original MC restricted to $\biguplus_{k=1}^{k0} S_k$.

In order to structurally bound the rates, different solutions have been proposed: the simplest solution [12] consists to take $\lambda^+_{i,j} = \max_{s' \in S_i} \left\{ \Sigma_{s" \in S_j}\ q_{s',s"} \right\}$ and $\mu^-_i = \min_{s' \in S_i} \left\{ \Sigma_{s" \in S_{i-1}}\ q_{s',s"} \right\}$. Such a solution has a serious drawback: in numerous models $\mu^-_i = 0$ which forbids the use of the method. Alternatively in particular cases [10], analytical expressions can be found for a general lower bound (see equation 6) of μ^-_i, but the application area is still very limited. At last, Carrasco [2] has proposed a general solution when activities inside the S_k subsets correspond to special phase-type distributions. The interest of the latter approach is twofold : it covers realistic applications and the computation of bounding rates is straightforward. However due to drastic simplifications, the lower bounds can be really far from the exact values.

Given a subset of states C of a MC and a reward r on the states of this MC, the aim of the present work is to bound the value of the reward on C averaged by the steady-state probability: $R = \pi(C)^{-1} \cdot \Sigma_{s \in C} \pi(s) \cdot r(s)$. It is now obvious that (for instance) taking $C = S_k$ and r as the cumulated rate to S_{k-1}, we specialize

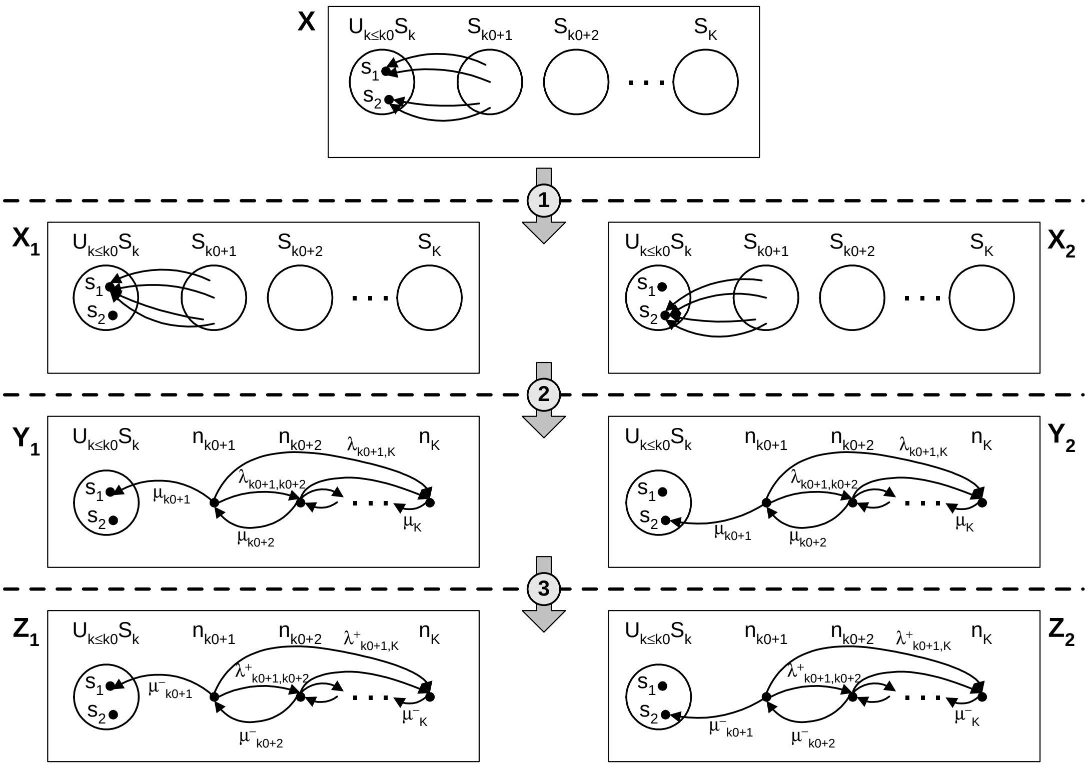

Figure 1: From bounds on rates to bounds on steady-state probabilities

our method to the previous case. The main hypotheses are that C can be partitioned into subsets C_i and that bounds on cumulated rates from a state to some C_i can be obtained by a structural analysis of the model. The examples of the paper will show that these are weak restrictions and moreover that often different partitions are possible where the choice of the appropriate partition is a trade-off between the accurateness of the bounds and the complexity of the computation.

More precisely, we present two algorithms. The first one applies to discrete-time Markov chains (DTMC) and is adaptable to continuous-time Markov chain (CTMC) via an “equivalent” uniformized discrete-time Markov chain. It relies on two key observations. The first observation is that the relative steady-state distribution of a subset of states depends on the steady-state distribution of entering the subset and the mean ratio of visits to a state before leaving the subset. This relation is such that without knowing the former quantity, one can still obtain bound with the latter one. The second observation is that this mean ratio can be lower bounded by the ratio between a lower bound of the mean number of the visits to a state and a upper bound of the mean number of the visits to all states. Whereas computing the lower bound presents no difficulty, the upper one requires the help of stochastic ordering theory [16]. Building on these observations, the algorithm is obtained by the following steps:

- Structural analysis of the model in order to obtain two sub-stochastic transitions matrices bounding the transition matrix restricted to C. This step is model-dependent and will be illustrated by the examples. The first one is a lower bound componentwise while the second one is a bound w.r.t. the strong ordering.

- Transformation of the second matrix into a monotonic one (which is still a bound of the original matrix) and verification that this new matrix is strictly sub-stochastic. We provide a weak and natural *necessary and sufficient condition* which can be easily checked *before* the transformation.

- Computation of two matrices of mean number of visits starting from the two bounding matrices.

- Computation of the bounds by a linear programming problem resolution.

Various algorithms have been developed in order to apply stochastic order theory to DTMCs ([8]) but to the best of our knowledge, none deals with sub-stochastic matrices. In a previous paper [5], we have proposed a similar algorithm. However it did not include the transformation step of the bounding matrix and consequently required that the original one is already monotonic thus restricting significantly the application area of the method.

It should be noted that in fact we have bounded the relative steady-state distribution independently of the type of the index r thus leading to a potential inaccuracy of the bound. The second algorithm applies to a CTMC for which the reward is the cumulated rate to another subset D of the CTMC (i.e. the introductory example). Here the key observation is that such a rate is the barycenter over states $s \in C$ of the ratio between the probability (starting from s) when leaving C to enter D and the sojourn time in C (starting from s). We will obtain a lower bound for each numerator and a upper bound for each denominator. Here again the difficult point is the computation of the upper bound that we solve with the help of the strong order.

The balance of the paper is the following one. In the second section we develop the first algorithm and we prove its correctness. The next section reports evaluations of the algorithm. In the forth section we develop the second algorithm and we prove its correctness. In the fifth section, we compare our bounds with the bound obtained by the method described in [2]. At last, we conclude and give some perspectives to this work.

2 Bounding expected steady-state reward rates in DTMCs

Hypotheses and notations

- In the DTMC $X = (E, \mathbf{P}_E)$, the state space E is partitioned as follows: $E = C \uplus D = \biguplus_{1 \leq i \leq n} C_i \uplus D$. The restriction of $\mathbf{P}_E$ to C is abbreviated by $\mathbf{P}$.
- We know upper and lower bounds for the reward function r on each C_i: $\forall i$, $\forall s \in C_i$, $r_i^- \leq r(s) \leq r_i^+$.
- We denote by $\widehat{X} = (\widehat{E}, \widehat{\mathbf{P}}_{\widehat{E}})$ the exact aggregation of X with $\widehat{E} = \{1, \ldots, n, n+1\}$, $I = \{1, \ldots, n\}$ corresponding to the subsets $C_1, \ldots, C_n$ and $n+1$ corresponding to D. The restriction of $\widehat{\mathbf{P}}_{\widehat{E}}$ to I is abbreviated by $\widehat{\mathbf{P}}$.
- We know componentwise lower and upper bounds of the matrix $\widehat{\mathbf{P}}$: $\widehat{\mathbf{P}}^-$ and $\widehat{\mathbf{P}}^+$.
- We know **out** a componentwise lower bound of the vector $\widehat{\mathbf{P}}_{\widehat{E}}[\cdot, n+1]$.

Objective

Computation of lower and upper bounds (R^- and R^+) of the Expected Steady-State Reward Rate (ESSRR) R conditioned on C: $R = \sum_{s \in C} r(s).\pi(s)/\pi(C)$.

2.1 Principle of the method

Reasoning at the aggregated level

We have $R = \sum_{s \in C} r(s).\pi(s)/\pi(C) = \sum_{i \in I} \pi(C_i)/\pi(C) \sum_{s \in C_i} r(s).\pi(s)/\pi(C_i)$. Thus denoting $\widehat{\pi}_i = \pi(C_i)/\pi(C)$ the conditional (on I) steady-state probability for $\widehat{X}$, we have:

$$\sum_{1 \leq i \leq n} r_i^- \widehat{\pi}_i = \sum_{1 \leq i \leq n} r_i^- \pi(C_i)/\pi(C) \leq R \leq \sum_{1 \leq i \leq n} r_i^+ \pi(C_i)/\pi(C) = \sum_{1 \leq i \leq n} r_i^+ \widehat{\pi}_i \quad (1)$$

Reducing the entries into I to a single point

Let us define $\widehat{X}^{(i_0)}$ the DTMC obtained by redirecting in $\widehat{X}$ all the outgoing arcs of $n+1$ towards $i_0 \in I$. It is well-known that denoting α the steady-state vector probability of entering I, we have: $\widehat{\pi}_i = \sum_{i_0 \in I} \alpha_{i_0} \widehat{\pi}_i^{i_0}$. Thus:

$$\min_{i_0 \in I} \left\{ \sum_{1\leq i\leq n} r_i^- \widehat{\pi}_i^{i_0} \right\} \leq \sum_{1\leq i\leq n} r_i^- \widehat{\pi}_i \leq R \leq \sum_{1\leq i\leq n} r_i^+ \widehat{\pi}_i \leq \max_{i_0 \in I} \left\{ \sum_{1\leq i\leq n} r_i^+ \widehat{\pi}_i^{i_0} \right\} \quad (2)$$

Exhibiting an explicit dependency of the expression on the transition matrix

We now focus on lower bounding $R^-(i_0) = \sum_{1\leq i\leq n} r_i^- \widehat{\pi}_i^{i_0}$ and upper bounding $R^+(i_0) = \sum_{1\leq i\leq n} r_i^+ \widehat{\pi}_i^{i_0}$. Let us denote $V(i,j)$ the mean number of visits to the state j starting from the state i before leaving the subset of states I. Due to the structure of $\widehat{X}^{(i_0)}$, we have:

$$\widehat{\pi}_i^{i_0} = \frac{V(i_0,i)}{\sum_{i' \in I} V(i_0,i')} = \frac{\left(\sum_{k\geq 0} \widehat{\mathbf{P}}^k\right)[i_0,i]}{\sum_{i'=1}^{n} \left(\sum_{k\geq 0} \widehat{\mathbf{P}}^k\right)[i_0,i']} \quad (3)$$

We look for a lower bound of $\widehat{\pi}_i^{i_0}$ i.e. for a lower bound of the numerator and a upper bound of the denominator in the equation 3.

Obtaining lower and upper bounds of the mean number of visits

Let us recall that a sub-stochastic matrix $\mathbf{M}$ is strictly sub-stochastic iff the infinite sum $\sum_{k\geq 0} \mathbf{M}^k$ is convergent. Since we know that $\widehat{\mathbf{P}}^-$ is a componentwise lower bound of $\widehat{\mathbf{P}}$, we have:

$$\sum_{k\geq 0} (\widehat{\mathbf{P}}^-)^k[i,j] \leq \sum_{k\geq 0} \widehat{\mathbf{P}}^k[i,j] \quad (4)$$

Obviously such an approach can not be applied to upper bound the denominator since the matrix $\widehat{\mathbf{P}}^+$ is generally not strictly sub-stochastic. However we note that we only need to upper bound the number of visits to the whole set I. The following definition and theorem[16] give us a way to obtain such an upper bound.

Definition 2.1 (Stochastic strong order notions) *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be* $n \times n$ *matrices with positive coefficients,*

- $\mathbf{A}$ *is* monotonic *iff* $\forall 1 \leq i < j \leq n,\ \forall 1 \leq m \leq n,\ \ \sum_{l=1}^{m} \mathbf{A}[i,l] \geq \sum_{l=1}^{m} \mathbf{A}[j,l]$
- $\mathbf{B}$ *is an* adapted bound *of* $\mathbf{A}$ *iff* $\forall 1 \leq i \leq n,\ \forall 1 \leq m \leq n,\ \sum_{l=1}^{m} \mathbf{B}[i,l] \geq \sum_{l=1}^{m} \mathbf{A}[i,l]$

Theorem 2.2 *Let a monotonic matrix* $\mathbf{B}$ *be an adapted bound of* $\mathbf{A}$*. Then:* $\forall k \in \mathbb{N}$, $\mathbf{B}^k$ *is monotonic and is an adapted bound of* $\mathbf{A}^k$.

In fact, an adapted bound $\mathbf{B}$ is nothing else than a $\leq_{st}$-lower bound of $\mathbf{A}$. However we prefer to use the word “adapted” in order to avoid a confusion since we use this bound to deduce an upper bound on the number of visits.

Based on this result, the core of our method is to obtain a monotonic adapted bound $\widehat{\mathbf{P}}^{um}$ of $\widehat{\mathbf{P}}$ from which the denominator of the expression in equation 3 can be bounded (by setting $m = n$ in the equation 5). The following equation states the nature of the bounds derived from this matrix.

$$\sum_{j=1}^{m} \sum_{k\geq 0} (\widehat{\mathbf{P}}^{um})^k[i,j] \geq \sum_{j=1}^{m} \sum_{k\geq 0} \widehat{\mathbf{P}}^k[i,j] \quad (5)$$

It should be noted that this kind of application of strong ordering is only significant for sub-stochastic matrices since it is a tautology in case of stochastic matrices. Once $\widehat{\mathbf{P}}^{um}$ is computed we need to check whether it is strictly sub-stochastic. Fortunately we have characterized the inputs for which this matrix is strictly sub-stochastic *without having to build it*. Thus this test is done a priori. We devote the next sub-section to a detailed presentation of the computation of $\widehat{\mathbf{P}}^{um}$ and a theoretical development about different strict sub-stochasticity criteria.

Algorithm 1 - Compute R^-, R^+ (Discrete time case)

Input: **out** a n vector, $\widehat{\mathbf{P}}^-,\widehat{\mathbf{P}}^+$: $n\times n$ matrices
begin
  Check the applicability of the method
   (i.e. whether $\widehat{\mathbf{P}}^{um}$ to be built will be strictly sub-stochastic)
  if not applicable **then** halt
  compute $\widehat{\mathbf{P}}^{um}$, a monotonic $\leq_{st}$-lower bound of $\widehat{\mathbf{P}}$ with $\widehat{\mathbf{P}}^-,\widehat{\mathbf{P}}^+$, **out**
  compute $\mathbf{V}^- = \left(\sum_{k\geq 0}(\widehat{\mathbf{P}}^-)^k\right)$ //lower bounds of the mean visit counts:
                  // $\mathbf{V}^-[i,j] \leq \mathbf{V}[i,j]$
  compute $\mathbf{V}^+ = \left(\sum_{k\geq 0}(\widehat{\mathbf{P}}^{um})^k\right)$ //upper bounds of the mean visit counts:
                  // $\forall j,m : \sum_{j\leq m}\mathbf{V}(i,j) \leq \sum_{j\leq m}\mathbf{V}^+(i,j)$
  //compute R^- and R^+ with $\mathbf{V}^-$ and $\mathbf{V}^+$:
  $R^- \leftarrow +\infty$; $R^+ \leftarrow 0$
  for $i_0 \leftarrow 1$ **to** n **do**
    $\widehat{\boldsymbol{\pi}}^{i_0-} \leftarrow \frac{1}{\mathbf{V}^+[i_0,.].\mathbf{1}_n}.\mathbf{V}^-[i_0,.]$ //lower bound of the distribution $\boldsymbol{\pi}^{i_0}$
    compute $R^-(i_0)$ and $R^+(i_0)$ as solutions of linear programming problems
    $R^- \leftarrow \min\{R^-,R^-(i_0)\}$; $R^+ \leftarrow \max\{R^+,R^+(i_0)\}$
  endfor
end

Finding the bounds $R^-(i_0)$ an $R^+(i_0)$ with linear programming
Once the lower bounds $\widehat{\boldsymbol{\pi}}^{i_0-}$ are computed, a straightforward bound of $R^-(i_0)$ is given by $\sum_{1\leq i\leq n} r_i^-.\widehat{\pi}_i^{i_0-}$, However a (usually) much better bound is obtained as the value of the objective function Z for the solution of the linear program:

$$\left|\begin{array}{ll} \min & Z = \sum_{1\leq i\leq n} r_i^-.x_i \\ \text{subject to} & \\ & \forall i\ \widehat{\pi}_i^{i_0-} \leq x_i \\ & \sum_{1\leq i\leq n} x_i = 1 \end{array}\right.$$

Here the increasing of the bound is due to the last equation stating that $\widehat{\boldsymbol{\pi}}^{i_0}$ is a probability distribution. Similarly, we obtain $R^+(i_0)$ as the value of the objective function Z for the solution of the linear program:

$$\left|\begin{array}{ll} \max & Z = \sum_{1\leq i\leq n} r_i^+.x_i \\ \text{subject to} & \\ & \forall i\ \widehat{\pi}_i^{i_0-} \leq x_i \\ & \sum_{1\leq i\leq n} x_i = 1 \end{array}\right. .$$

Let us recall that the time complexity of the resolution of such problems is polynomial w.r.t. n [13].

Algorithm 1 summarizes our method for DTMCs. There are different numerical methods to compute the infinite sums (which are indeed inversions). A possible way is to truncate the sums to keep the error at a given level [15].

Finally let us discuss about the sources of inaccuracy of the computed bounds. They are either related to the partition of $C = \biguplus_{1\leq i\leq n} C_i$ or related to the interaction between C and D.

1. A set C_i must be quite homogeneous w.r.t. the reward in order to keep r_i^- and r_i^+ as close as possible.

2. A set C_i must be quite homogeneous w.r.t. the probability to jump into another set C_j or into D in order to keep $\widehat{\mathbf{P}}^-$, $\widehat{\mathbf{P}}^+$ and **out** as close to the exact values as possible.

3. The order of the items in I should be appropriately chosen in order that the monotonicity requirement only leads to minor transformations of the intermediate adapted bound of $\widehat{\mathbf{P}}$ (see the next

Algorithm 2 - Build $\widehat{\mathbf{P}}^u$

Input: **out** a n vector, $\widehat{\mathbf{P}}^-, \widehat{\mathbf{P}}^+$: $n \times n$ matrices
begin
  for $i \leftarrow 1$ **to** n **do**
    // compute the "accumulation" matrix $\widehat{\mathbf{P}}^{acc}$:
    for $j \leftarrow 1$ **to** n **do**
      $\widehat{\mathbf{P}}^{acc}[i,j] \leftarrow \min\left(\sum_{k=1}^{j} \widehat{\mathbf{P}}^+[i,k], 1 - \sum_{k=j+1}^{n} \widehat{\mathbf{P}}^-[i,k] - \mathbf{out}[i]\right)$
      (Ih) **if** $i = j$ **and** $\widehat{\mathbf{P}}^{acc}[i,j] = 1$ **then** halt
    endfor
    // compute the adapted matrix $\widehat{\mathbf{P}}^u$:
    $\widehat{\mathbf{P}}^u[i,1] \leftarrow \widehat{\mathbf{P}}^{acc}[i,1]$
    for $j \leftarrow 2$ **to** n **do**
      $\widehat{\mathbf{P}}^u[i,j] \leftarrow \widehat{\mathbf{P}}^{acc}[i,j] - \widehat{\mathbf{P}}^{acc}[i,j-1]$
    endfor
  endfor
end

sub-section).

4. At last, the redirections of $\widehat{X}$ towards a single entry point should not alter too much the averaged value of the reward.

2.2 Computing the monotonic adapted bound $\widehat{\mathbf{P}}^{um}$

We proceed in two steps. At first, starting with $\widehat{\mathbf{P}}^-$, $\widehat{\mathbf{P}}^+$ and **out** we build $\widehat{\mathbf{P}}^u$ an adapted bound of $\widehat{\mathbf{P}}$. This matrix is sub-stochastic. Then we apply a transformation on this matrix in order to make it monotonic.

2.2.1 Computing the adapted bound $\widehat{\mathbf{P}}^u$

Algorithm 2 first builds a matrix $\widehat{\mathbf{P}}^{acc}$ where $\widehat{\mathbf{P}}^{acc}[i,j]$ is the best computable upper bound of $\Sigma_{k=1}^{j}\widehat{\mathbf{P}}[i,k]$ w.r.t. the inputs. Due to the second component of the *min*, $\widehat{\mathbf{P}}^{acc}[i,n] \leq 1$. Then by "differentiating" this matrix we obtain $\widehat{\mathbf{P}}^u$ which is a sub-stochastic adapted upper bound of $\widehat{\mathbf{P}}$. The meaning of the statement (Ih) will be explained in the handling of the strict sub-stochasticity.

For instance, if $\widehat{\mathbf{P}}^+[i,.] = [0.2,\ 0.4,\ 0.5]$, $\widehat{\mathbf{P}}^-[i,.] = [0.1,\ 0.15,\ 0.3]$ and $\mathbf{out}[i] = 0.2$ ($n = 3$), we obtain $\widehat{\mathbf{P}}^u[i,.] = [0.2,\ 0.15,\ 0.45]$.

2.2.2 Building a monotonic matrix $\widehat{\mathbf{P}}^{um}$ from $\widehat{\mathbf{P}}^u$

In the general case, $\widehat{\mathbf{P}}^u$ built above is not monotonic. So we now build a monotonic version of $\widehat{\mathbf{P}}^u$, denoted by $\widehat{\mathbf{P}}^{um}$ such that equation 5 holds. Algorithm 3 is a "sub-stochastic version" of the one proposed by [17, 1]. It is based on the lemma 2.3 below.

Lemma 2.3 *At the i^{th} iteration of the outer loop and at the beginning of the j^{th} iteration of the inner loop of algorithm 3, one has the following equalities. These equalities also hold for $j = n+1$ with the meaning that the program is exiting the inner loop.*

1. $y = \Sigma_{k=1}^{j-1}\widehat{\mathbf{P}}^{um}[i+1,k]$

2. $x = \max\left(\Sigma_{k=1}^{j-1}\widehat{\mathbf{P}}^u[i,k], y\right) = \Sigma_{k=1}^{j-1}\widehat{\mathbf{P}}^{um}[i,k]$

Algorithm 3 - Build a monotonic version of an adapted bound

Input: $\widehat{\mathbf{P}}^u$: a $n \times n$ adapted bounding matrix
begin
$\quad \widehat{\mathbf{P}}^{um}[n,\cdot] \leftarrow \widehat{\mathbf{P}}^u[n,\cdot]$
$\quad$ **for** $i \leftarrow n-1$ **downto** 1 **do**
$\quad\quad$ (I0) $x \leftarrow 0; y \leftarrow 0; \Delta \leftarrow 0$
$\quad\quad$ **for** $j \leftarrow 1$ **to** n **do**
$\quad\quad\quad$ (I1) $\widehat{\mathbf{P}}^{um}[i,j] \leftarrow \max\left(\widehat{\mathbf{P}}^u[i,j]-\Delta, y+\widehat{\mathbf{P}}^{um}[i+1,j]-x\right)$
$\quad\quad\quad$ (I2) $y \leftarrow y+\widehat{\mathbf{P}}^{um}[i+1,j]$
$\quad\quad\quad$ (I3) $x \leftarrow x+\widehat{\mathbf{P}}^{um}[i,j]$
$\quad\quad\quad$ (I4) $\Delta \leftarrow \Delta+\widehat{\mathbf{P}}^{um}[i,j]-\widehat{\mathbf{P}}^u[i,j]$
$\quad\quad$ **endfor**
$\quad$ **endfor**
end

---

3. $\Delta = x - \Sigma_{k=1}^{j-1}\widehat{\mathbf{P}}^u[i,k]$

Furthermore the item $\widehat{\mathbf{P}}^{um}[i,j]$ *will take a positive value during the execution of the next statement.*

Proof.
We prove the lemma by induction on j.
For $j=1$, the three equalities are due to the instruction (I0).
Let us suppose that we have proven the lemma until some value j. In order to analyze the effect of instruction (I1), we substitute to y and Δ the right-hand side of the equalities. This gives us:
$\widehat{\mathbf{P}}^{um}[i,j] \leftarrow \max\left(\Sigma_{k=1}^{j}\widehat{\mathbf{P}}^u[i,k]-x, \Sigma_{k=1}^{j}\widehat{\mathbf{P}}^{um}[i+1,k]-x\right) =$
$\max\left(\Sigma_{k=1}^{j}\widehat{\mathbf{P}}^u[i,k], \Sigma_{k=1}^{j}\widehat{\mathbf{P}}^{um}[i+1,k]\right) - x \geq 0$
The latter inequality is due to the first expression of x. The equality (1) is inductively proved due to the instruction (I2). Let us analyze the new value of x taken after the instruction (I3). We substitute to $\widehat{\mathbf{P}}^{um}[i,j]$ the expression we have obtained in our previous analysis. This gives us:
$x \leftarrow \max\left(\Sigma_{k=1}^{j}\widehat{\mathbf{P}}^u[i,k], \Sigma_{k=1}^{j}\widehat{\mathbf{P}}^{um}[i+1,k]\right)$
which is exactly the first expression of x. The second one is inductively proved by a simple examination of (I3).
Now we analyze the value taken by Δ during the execution of instruction (I4). We substitute the old value of Δ by the expression of the third equality which gives us:
$\Delta \leftarrow x - \Sigma_{k=1}^{j-1}\widehat{\mathbf{P}}^u[i,k] + \widehat{\mathbf{P}}^{um}[i,j] - \widehat{\mathbf{P}}^u[i,j] = \Sigma_{k=1}^{j-1}\widehat{\mathbf{P}}^{um}[i,k] - \Sigma_{k=1}^{j-1}\widehat{\mathbf{P}}^u[i,k] + \widehat{\mathbf{P}}^{um}[i,j] - \widehat{\mathbf{P}}^u[i,j]$.
The last equality has been obtained by the second expression of x in the second equality. $\diamond$

At first, the algorithm sets the last row of the new matrix to the same row of the old one. Then it sets each other row in decreasing order in such a way that a partial sum of the row is the least upper bound of the corresponding sum of the original matrix and the corresponding partial sum of the next row (which has already been set). The variable Δ of the algorithm represents the excess of the current partial sum w.r.t. the original partial sum. The previous lemma (more precisely the second equality and the last assertion) proves that the transformed matrix is the minimal one satisfying the required property. It is straightforward that this new matrix is still a sub-stochastic matrix. This algorithm could be implemented with a single matrix but we have chosen the current presentation in order to simplify the proof.

2.2.3 Checking and characterizing the strict sub-stochasticity of $\widehat{\mathbf{P}}^{um}$

Given a sub-stochastic matrix $\widehat{\mathbf{P}}^{um}$, one checks in linear time w.r.t. the number of non null items of the matrix whether it is strictly sub-stochastic with the following procedure:

1. Define a oriented graph where the set of nodes is $I \cup \{n+1\}$
2. There is an arc between $i \in I$ and $j \in I$ iff $\widehat{\mathbf{P}}^{um}[i,j] \neq 0$
3. There is an arc between $i \in I$ and $n+1$ iff $\Sigma_{j=1}^{n} \widehat{\mathbf{P}}^{um}[i,j] \neq 1$
4. Check whether $n+1$ is reachable from any node. This can be done by a breadth-first backward search starting from $n+1$.

The interested reader will find in [6] the proof of the correctness of this algorithm. The important point here is that the strict sub-stochasticity depends on structural criterion: whether an item is null and whether a row sum is 1. It would be interesting to have a similar structural characterization depending on $\widehat{\mathbf{P}}^{+}$, $\widehat{\mathbf{P}}^{-}$ and **out** as it would give insight on which kind of rates bounds could be handled by our method. This is the goal of next lemma.

Lemma 2.4 *The following assertions are equivalent:*

1. $\widehat{\mathbf{P}}^{um}$ *is strictly sub-stochastic*

2. $\forall i \; \sum_{j\leq i} \widehat{\mathbf{P}}^{u}[i,j] < 1$

3. $\forall i \; \sum_{j\leq i} \widehat{\mathbf{P}}^{+}[i,j] < 1$ *or* $\sum_{j>i} \widehat{\mathbf{P}}^{-}[i,j] + \mathbf{out}[i] > 0$

Proof.
The assertions 2 and 3 are equivalent due to the construction of $\widehat{\mathbf{P}}^{u}$. Thus we will prove the equivalence of 1 and 2.

At first, let us suppose that the assertion 2 is satisfied. We claim that the same condition is satisfied for $\widehat{\mathbf{P}}^{um}$. We prove it by a reverse induction on i. If $i = n$ then it is immediate since the last rows of the two matrices are identical. Let us suppose that the inequalities are satisfied for the rows $k > i$. Then we know (see lemma 2.3):
$\sum_{j\leq i} \widehat{\mathbf{P}}^{um}[i,j] = \max\left(\sum_{j\leq i} \widehat{\mathbf{P}}^{u}[i,j], \sum_{j\leq i} \widehat{\mathbf{P}}^{um}[i+1,j]\right) \leq$
$\max\left(\sum_{j\leq i} \widehat{\mathbf{P}}^{u}[i,j], \sum_{j\leq i+1} \widehat{\mathbf{P}}^{um}[i+1,j]\right) < 1$
Thus in the graph associated to $\widehat{\mathbf{P}}^{um}$:

- either $\sum_{j>i} \widehat{\mathbf{P}}^{um}[i,j] = 0$ and there is an arc from i to $n+1$
- or $\sum_{j>i} \widehat{\mathbf{P}}^{um}[i,j] > 0$ and there is an arc from i to $j > i$

So starting from any node i and following these arcs the node $n+1$ will be eventually reached.

Now suppose that the assertion 2 is not satisfied i.e.; $\exists i \; \sum_{j\leq i} \widehat{\mathbf{P}}^{u}[i,j] = 1$. Thus since $\widehat{\mathbf{P}}^{um}$ is an adapted bound of $\widehat{\mathbf{P}}^{u}$, one has $\sum_{j\leq i} \widehat{\mathbf{P}}^{um}[i,j] = 1$ and since $\widehat{\mathbf{P}}^{um}$ is monotonic, one has $\forall k \leq i \sum_{j\leq i} \widehat{\mathbf{P}}^{um}[k,j] = 1$ but this means that in the associated graph of the above procedure, the subset of nodes $\{1,\ldots,i\}$ has no outgoing arc. Then $n+1$ is unreachable from this subset of states. ♢

Thus we directly check on the inputs whether our method is applicable. This is done without extra-computation by the statement (Ih) of algorithm 2 which builds $\widehat{\mathbf{P}}^{u}$. Roughly speaking the criterion means that in the system, for any C_i there is either a C_j with $j > i$ which any state of C_i can enter or any state of C_i can exit C. More informally, the criterion states that if the order of the indices i is related to some progress measure then the system has always a non null probability to progress. Of course, this does not preclude the probability of “regression” (which was forbidden by our previous algorithm [5]).

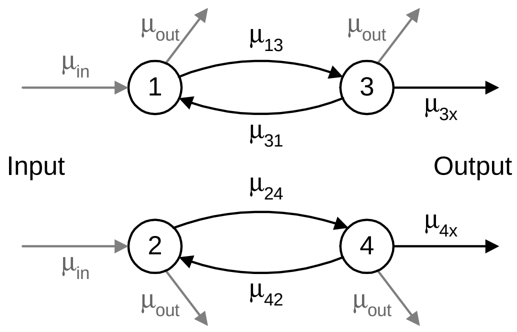

Figure 2: A simple four stages service

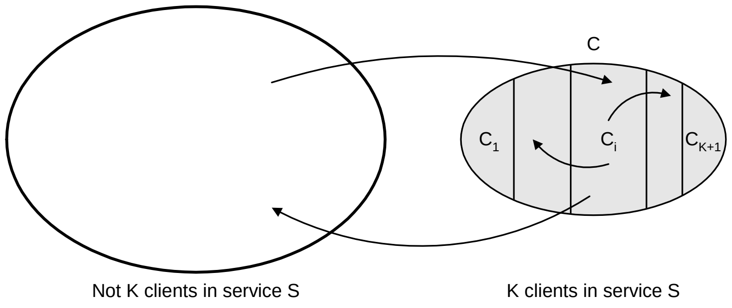

Figure 3: State space of the system

3 Example and numerical results

We present below an example of application of our approach and we analyze the numerical results. All the computations were done with the help of the Scilab numerical environment (see `http://scilabsoft.inria.fr/`).

3.1 The model

We consider a system with several entities (clients). Each client may execute various activities. One activity consists of requiring a service, say S. S is made of a network of four exponential stages (figure 2). During the execution of S, the system may execute some action (with rate μ_{out}) that enforces some clients (if any) to leave S: this could be the case if for instance the service policy is preemptive for all stages of S. Some clients may also begin a S service with (cumulate) rate μ_{in}. When a client is in stage i, it accumulates a reward $r(i)$ during one time unit. We apply our method to compute bounds of the conditional reward R when there are K clients running S.

In this example, C is the set of global states where K clients are served and the state space of the system is depicted in figure 3. A Markovian state s of C is $s = (s_0, k_1, k_2, k_3, k_4)$ with $\sum_i k_i = K$. k_i is the number of clients in stage i and s_0 describes the state of the rest of the system when K clients are in the service S. We have $|C| = |C_0|.\binom{K+4-1}{4-1} = \frac{(K+3)(K+2)(K+1)}{6} \simeq |C_0|.K^3$ where $|C_0|$ is the number of states s_0.

If we assume that r approximatively increases (or decreases) when the service progresses, we choose the partition $C = \biguplus_{1\leq i\leq K+1} C_i$ with $C_i = \{s |\ k_1 + k_2 = K - i + 1, k_3 + k_4 = i - 1\}$. Doing so, we group all states with the same number of clients in the stages $1,2$ and $3,4$. The cardinality of each C_i is greater than $(K - i + 2)i$ and we have $K + 1$ aggregates C_i. The reduction of cardinality between C and the aggregated state space is then $O(K^2)$.

Choosing an order among the C_i is very important and must take into account the behaviour of the system. In our case, the variation of r and the structure of the service S lead us to order the subsets by their index i, i.e. $C_i \leq C_{i'}$ iff $i \leq i'$.

We can define the $\widehat{\mathbf{P}}^-$ and $\widehat{\mathbf{P}}^+$ matrices as follows. First we compute an upper bound m of the sum of the rates in all stages to uniformize the CTMC:

$$m = 1.1\,(K \max\{\mu_{13}, \mu_{24}, \mu_{31}, \mu_{42}, \mu_{3x}, \mu_{4x}\} + \max\{\mu_{out} + \mu_{in}\})$$

$\widehat{\mathbf{P}}^+$ is then a band matrix with only under-diagonal, diagonal and upper-diagonal non null terms:

	1 $(K,0)$	$\ldots$	$i-1$ $(K-i+2,i-2)$	i $(K-i+1,i-1)$	$i+1$ $(K-i,i)$	$\ldots$	$K+1$ $(0,K)$
1 $(K,0)$	$*$						
$\vdots$		$\ddots$					
$i-1$ $(K-i+2,i-2)$			$*$	$*$			
i $(K-i+1,i-1)$			$n_2.\max(\mu_{31},\mu_{42})/m$	$\widehat{\mathbf{P}}^+[i,i]$	$n_1.\max(\mu_{13},\mu_{24})/m$		
$i+1$ $(K-i,i)$				$*$	$*$		
$\vdots$						$\ddots$	
$K+1$ $(0,K)$							$*$

The diagonal terms of $\widehat{\mathbf{P}}^+$ are

$$\widehat{\mathbf{P}}^+[i,i] = 1 - \sum_{j \neq i} \widehat{\mathbf{P}}^-[i,j] - \frac{n_2.\min(\mu_{3x}, \mu_{4x}) + \min(\mu_{in} + \mu_{out})}{m}$$

meaning that we may leave i either by moving inside I or by leaving C due either to an end of the service S (μ_{3x}, μ_{4x}) or due to an external event (μ_{out} or μ_{in}). The $\widehat{\mathbf{P}}^-$ matrix has the same structure with $\widehat{\mathbf{P}}^-[i,i-1] = n_2.\min(\mu_{31},\mu_{42})/m$, $\widehat{\mathbf{P}}^-[i,i] = 1 - \sum_{j \neq i} \widehat{\mathbf{P}}^+[i,j] - \frac{n_2.\max(\mu_{3x},\mu_{4x}) + \max(\mu_{in}+\mu_{out})}{m}$ and $\widehat{\mathbf{P}}^-[i,i+1] = n_1.\min(\mu_{13},\mu_{24})/m$.

Finally we set $\mathbf{out}[i] = \frac{n_2.\min(\mu_{3x},\mu_{4x}) + \min(\mu_{in}+\mu_{out})}{m}$.

3.2 Numerical results

We have first computed our bounds for a "decreasing" lower bound of the reward r given by the decreasing function: $r^-(i) = r_0 + (n-i).T$, with $T = 100$. Since r_0 is not a significant parameter with respect to the quality of the bounds, we set $r_0 = 0$ in the sequel.

In order to evaluate the behaviour of our bounds, we introduce two uncertainty parameters: Δ_r ($0 \leq \Delta_r$) is related to the knowledge on the reward r on C, and Δ_μ ($0 \leq \Delta_\mu$) is used to set the ratio between the min and max rate values.

More precisely, the uncertainty on r is described by Δ_r such that $\forall\, i$, $r^+(i) = (1+\Delta_r)r^-(i)$ and $\forall s \in C_i$, $r^-(i) \leq r(s) \leq r^+(i)$. For what concerns the stochastic parameters we choose: $\mu_{13} = 1$, $\mu_{31} = 0.1$, $\mu_{3x} = 1$ and $\min\{\mu_{in} + \mu_{out}\} = 0.01$. The values μ_{24}, μ_{40} and $\max\{\mu_{in} + \mu_{out}\}$ are chosen as $(1+\Delta_\mu)$ times the previous values. Hence, $\min\{\mu_{13},\mu_{24}\} = \mu_{13}$, $\max\{\mu_{13},\mu_{24}\} = \mu_{24} = (1+\Delta_\mu)\mu_{13}$ and so on for the other couples of rates. Note that for these realistic values of parameters $\widehat{\mathbf{P}}^+$ is often (depending on Δ_μ) monotonic, so that our algorithm will set $\widehat{\mathbf{P}}^{um} = \widehat{\mathbf{P}}^u$. Moreover $\widehat{\mathbf{P}}^+$ is never sub-stochastic, due to our definition of its diagonal term. This exemplifies the interest of the intermediate matrix $\widehat{\mathbf{P}}^{acc}$ (algorithm 2).

The experiments were conducted for $K = 10$ unless otherwise stated.

In order to appreciate the quality of our bounds, we computed a "pseudo" aggregated DTMC X with state space $I \bigcup \{n+1\}$ and matrix $\widehat{\mathbf{P}}^X$ such that upper-left $n \times (n+1)$ block of $\widehat{\mathbf{P}}^X$ is $(\widehat{\mathbf{P}}^+ + \widehat{\mathbf{P}}^-)/2$. This DTMC may be seen as a "mean value" between the two bounds $\widehat{\mathbf{P}}^-$ and $\widehat{\mathbf{P}}^+$. However, the unknown and important values are the steady-state transition probabilities to enter I. Choosing these n values (i.e.; defining the row $\widehat{\mathbf{P}}^X[n+1,.]$) allows us to consider all possibilities: for instance with $\widehat{\mathbf{P}}^X[n+1,1] = 1.0$, the unique entry point of I is the first aggregate. For each set of parameters, we used several possibilities

Δ_r (%)	Δ_μ (%)	$\Delta^n(R^-)$ (%)	R^-	R^n, R^u, R^1	R^+	$\Delta^1(R^+)$ (%)	$\Delta_{R^+}(m_r)$ (%)	$m_r = \max\{r(i)\}$
0	0	0	10	10,544,788	788	0	27	1000
⋮	10	-19	8	⋮	864	10	16	⋮
⋮	20	-33	7	⋮	906	15	10	⋮
⋮	50	-59	4	⋮	961	22	4	⋮
10	0	-4	10	11,571,827	867	5	27	1100
⋮	10	-23	8	⋮	950	15	16	⋮
⋮	20	-36	7	⋮	997	21	10	⋮
⋮	50	-61	4	⋮	1057	28	4	⋮
20	0	-9	10	11,597,867	945	9	27	1200
⋮	10	-27	8	⋮	1036	20	16	⋮
⋮	20	-40	7	⋮	1088	26	10	⋮
⋮	50	-63	4	⋮	1153	33	4	⋮

Table 1: Bounds for $k = 10$ (decreasing reward)

Δ_r (%)	Δ_μ (%)	$\Delta^n(R^-)$ (%)	R^-	R^n, R^u, R^3	R^+	$\Delta^3(R^+)$ (%)	$\Delta_{R^+}(m_r)$ (%)	$m_r = \max\{r(i)\}$
0	0	0	10	10,466,661	661	0	51	1000
⋮	10	-19	8	⋮	744	13	34	⋮
⋮	20	-33	7	⋮	798	21	25	⋮
⋮	50	-59	4	⋮	892	34	13	⋮

Table 2: Bounds for $k = 10$ (perturbed reward r)

with respect to the entry points in I: in 1 (1), in 3 (3), in $n = K+1$ (n), uniformly (U) on I. Using the reward r, given by $\forall i,\ \forall s \in C_i,\ r(s) = (r^-(i) + r^+(i))/2$, we have computed the mean rewards R^1, R^3, R^u and R^n, given in the tables below.

We first compare our bounds with the minimal and maximal values of r on C, immediately available. We have $\min_{s\in C} r(s) = 0$. In our experiments, R^- varies from 10 to 212. In any case, our method provides a significant non zero lower bound. The ratio $\Delta_{R^+}(m_r) = \frac{\max\{r(s)\} - R^+}{R^+}$ is given in the tables below. Depending on the parameters, $\Delta_{R^+}(m_r)$ varies from a few percents (not very significant) to 51% (important improvement).

The second criterion is given by the difference between R^+ and R^- and the extremal values R^1, R^3 and R^n reachable by X (see above). The results are summarized in the tables 1, 2, 3, 4. In these tables, $\Delta^x(R^-) = \frac{R^x - R^-}{R^x}$ gives the relative deviation of R^- w.r.t.; R^x and $\Delta^x(R^+) = \frac{R^+ - R^x}{R^x}$ represents the relative deviation of R^+ w.r.t.; R^x.

With a decreasing reward r^- (table 1), we observe

- a low sensibility of the bounds to the variation (Δ_r) of r,
- a high sensibility of the bounds to the variation (Δ_μ) of the rates. When Δ_μ increases, the quality of R^- first rapidly decreases but then is almost sub-linear with Δ_μ. The variation of the quality of R^+ is always sub-linear.

Δ_r (%)	Δ_μ (%)	$\Delta^1(R^-)$ (%)	R^-	R^1, R^u, R^n	R^+	$\Delta^n(R^+)$ (%)	$\Delta_{R^+}(m_r)$ (%)	$m_r = \max\{r(i)\}$
0	0	0	212	212,456,989	989	0	1	1000
⋮	10	-35	136	⋮	991	13	1	⋮
⋮	20	-56	93	⋮	993	21	1	⋮
⋮	50	-82	39	⋮	995	34	0	⋮

Table 3: Bounds for $k = 10$ (increasing reward)

Δ_r (%)	Δ_μ (%)	$\Delta^n(R^-)$ (%)	R^-	R^n, R^u, R^3	R^+	$\Delta^3(R^+)$ (%)	$\Delta_{R^+}(m_r)$ (%)	$m_r = \max\{r(i)\}$
0	0	0	10	10,3143,4351	4351	0	15	5000
⋮	10	-19	8	⋮	4667	7	7	⋮
⋮	20	-33	7	⋮	4796	10	4	⋮
⋮	50	-59	4	⋮	4920	13	2	⋮

Table 4: Bounds for $k = 50$ (perturbed reward r)

It is worth noting that the bounds are very sensitive to the order of the aggregates and the variation of r between these sets. We illustrate this phenomenon by two other experimentations. We first "perturb" the decrease of r: we set $r(2) = 1.0$ (instead of 900). The results are given in table 2. We observe that the quality of the bounds is quite stable, but also that the improvement of R^+ w.r.t. to $\max\{r(i)\} = m_r$ is significant (51% versus 27%, etc.). Another modification of the initial model is to set an *increasing* reward r. Results are reported in table 3 - note that R^1, R^u, R^n are given in this (reverse with respect to tables 1, 2) order. We can note that the upper bound R^+ is very close to the effective maximum (but note that this maximum is nearly the trivial bound $\max\{r(i)\} = m_r$). We also remark that R^- is less precise than in the decreasing case.

We also did several computations with different values of k. Table 4 gives the results for $k = 50$ and a "perturbed" ($r(2) = 1.0$) reward (compare with table 2). Results on R^- are identical for $k = 10, 50$ and results on R^+ are better for $k = 50$; note however that R^+ is closer to $\max\{r(i)\} = m_r$ for $k = 50$.

The previous sets of parameter values lead to a monotonic $\widehat{\mathbf{P}}^{um}$. In order to study the effect of monotonization, we have set $\mu_{31} = 1.0$ instead of 0.1, letting all other parameters unchanged. With this new value, $\widehat{\mathbf{P}}^{um}$ is no more monotonic. Results (to be compared with table 1) are given in table 5.

We first notice that the monotonization process is not the critical factor w.r.t. the quality of the bounds. Moreover the lower bound is still close to the minimal value relative to the pseudo-aggregated model and far from zero.

4 Lower bounding output rate in CTMCs

This section presents a method to bound the steady-state rate out of a subset C of states to another subset A of a CTMC whose state space is $E = C \uplus A \uplus B$. Note that reversing A and B, the same method could also provide an upper bound of this rate (see section 4.3).

Hypotheses and notations

- In the CTMC $X = (E, \mathbf{Q})$, the state space E is partitioned as follows: $E = C \uplus A \uplus B = \uplus_{1 \leq i \leq n} C_i \uplus A \uplus B$
- We denote by $\widehat{X} = (\widehat{E}, \widehat{\mathbf{Q}})$, the exact aggregation of X with $\widehat{E} = \{1, \ldots, n, n+1, n+2\}$ $I = \{1, \ldots, n\}$ corresponding to the subsets $C_1, \ldots, C_n$, $n+1$ corresponding to A and $n+2$ corresponding to B. We denote by $\widehat{\mathbf{P}}$ the transition probability matrix of the embedded DTMC of $\widehat{X}$

Δ_r (%)	Δ_μ (%)	$\Delta^n(R^-)$ (%)	R^-	R^n, R^u, R^1	R^+	$\Delta^1(R^+)$ (%)	$\Delta_{R^+}(m_r)$ (%)	$m_r = \max\{r(i)\}$
0	0	-20	87	109,598,808	834	3	20	1000
⋮	10	-47	57	⋮	901	11	11	⋮
⋮	20	-63	41	⋮	934	16	7	⋮
⋮	50	-83	19	⋮	974	21	3	⋮
10	0	-24	87	115,628,848	917	8	20	1100
⋮	10	-50	57	⋮	991	17	11	⋮
⋮	20	-64	41	⋮	1028	21	7	⋮
⋮	50	-83	19	⋮	1071	26	3	⋮
20	0	-28	87	120,657,888	1001	13	20	1200
⋮	10	-52	57	⋮	1080	22	11	⋮
⋮	20	-66	41	⋮	1121	26	7	⋮
⋮	50	-84	19	⋮	1169	32	3	⋮

Table 5: Bounds for $k = 10$ (non monotonic $\widehat{\mathbf{P}}^{um}$, decreasing case)

- We denote by $\widetilde{X} = (\widetilde{E}, \widetilde{\mathbf{Q}})$, the exact aggregation of X with $\widetilde{E} = \{0, n+1, n+2\}$ 0 corresponding to the subset C, $n+1$ corresponding to A and $n+2$ corresponding to B. Let us remark that this CTMC is also an exact aggregation of $\widehat{X}$. We denote by $\widetilde{\mathbf{P}}$ the transition probability matrix of the embedded DTMC of $\widetilde{X}$.
- We know a $n \times (n+1)$ matrix $\widehat{\mathbf{P}}^-$ such that $\widehat{\mathbf{P}}^-[i,j] \leq \widehat{\mathbf{P}}[i,j]$ for $1 \leq i \leq n$ and $1 \leq j \leq n+1$.
- We know a lower triangular $n \times n$ matrix $\widehat{\mathbf{Q}}^+$ such that $\widehat{\mathbf{Q}}[i,j] \leq \widehat{\mathbf{Q}}^+[i,j]$ for $1 \leq i \leq n$ and $1 \leq j < i$
- We know an upper triangular $n \times (n+1)$ matrix $\widehat{\mathbf{Q}}^-$ such that $\widehat{\mathbf{Q}}^-[i,j] \leq \widehat{\mathbf{Q}}[i,j]$ for $1 \leq i \leq n$ and $i < j \leq n$ and $\widehat{\mathbf{Q}}^-[i,n+1] \leq \widehat{\mathbf{Q}}[i,n+1] + \widehat{\mathbf{Q}}[i,n+2]$
- The restriction of a matrix $\mathbf{M}$ to a subset of items $J \times J$ will be denoted by $\mathbf{M}_{|J}$.

Objective
Computation of $\mu^{(A)-}$ a lower bound of $\mu^{(A)} = \sum_{s\in C}\sum_{s'\in A}\mathbf{Q}[s,s'].\pi(s)/\pi(C)$, the steady-state rate from C to A.

4.1 Principle of the method

Reasoning at the aggregated level
We have:

$$\mu^{(A)} = \sum_{s\in C}\sum_{s'\in A}\mathbf{Q}[s,s'].\pi(s)/\pi(C) = \widetilde{\mathbf{Q}}[0,n+1] = -\widetilde{\mathbf{P}}[0,n+1]\cdot\widetilde{\mathbf{Q}}[0,0]$$

$\widetilde{X}$ is an exact aggregation of $\widehat{X}$. Thus in the latter chain, $\widetilde{\mathbf{P}}[0,n+1]$ is the steady-state probability of reaching $n+1$ on leave from I and $-1/\widetilde{\mathbf{Q}}[0,0]$ is the steady-state holding time in I. Let us denote α the steady-state probability vector of entering I, $\widehat{p}_i^{(A)}$ the probability of reaching $n+1$ on leave from I when starting from i and $\widehat{h}_i$ the holding time in I when starting from i. The next equations relate the previous quantities (see for instance [14, 11]).

$$\widetilde{\mathbf{P}}[0,n+1] = \sum_{I\in I}\alpha_i\widehat{p}_i^{(A)} \qquad -1/\widetilde{\mathbf{Q}}[0,0] = \sum_{i\in I}\alpha_i\widehat{h}_i$$

We deduce that :

$$\widehat{\mu}^{(A)} = \frac{\sum_{i=1}^{n} \alpha_i \widehat{p}_i^{(A)}}{\sum_{i=1}^{n} \alpha_i \widehat{h}_i} = \sum_{i=1}^{n} \frac{\alpha_i \widehat{h}_i}{\sum_{i=1}^{n} \alpha_i \widehat{h}_i} (\widehat{p}_i^{(A)} / \widehat{h}_i) \geq \min_i \frac{\widehat{p}_i^{(A)}}{\widehat{h}_i} \tag{6}$$

So in the sequel, we focus on lower bounding $\widehat{p}_i^{(A)}$ and upper bounding $\widehat{h}_i$.

Exhibiting an explicit dependency of $\widehat{p}_i^{(A)}$ on the transition matrix $\widehat{\mathbf{P}}$

We obtain the probability $\widehat{p}_i^{(A)}$ to leave I for $n+1$, when starting from i by conditioning this probability on the number of transitions before leaving I. This gives:

$$\widehat{p}_i^{(A)} = \sum_{j \in I} \left(\sum_{k \geq 0} (\widehat{\mathbf{P}}_{|I})^k \right) [i, j] \times \widehat{\mathbf{P}}[j, n+1]$$

Since the right-hand side of this equation is composed by positive terms, sums and products, we only have to lower bound each item of this expression by the corresponding item of $\widehat{\mathbf{P}}^-$ in order to obtain a lower bound of $\widehat{p}_i^{(A)}$.

Interpreting $\widehat{h}_i$ as a transient measure

Let us define $\widehat{X}^a$ obtained from $\widehat{X}$ by merging the set of states $\{n+1, n+2\}$ into a single **absorbing** one $n+1$ corresponding to the subset $A \uplus B$. Let $\widehat{\boldsymbol{\pi}}_i^a(t)$ be the probability vector of $\widehat{X}^a$ at time t with initial distribution $\widehat{\boldsymbol{\pi}}_i^a(0) = \mathbf{1}_i$, i.e. $\widehat{X}^a$ starting from i. By definition of $\widehat{X}^a$, the holding time $\widehat{h}_i$ can be expressed as:

$$\widehat{h}_i = \int_0^{+\infty} \sum_{j \in I} \widehat{\pi}_i^a(t)[j] dt$$

We will exploit this expression with the help of the following theorem related to strong stochastic order in CTMCs (the conditions given are expressed differently in [16], Th. 4.2.8, p.67).

Theorem 4.1 ([16]) *Let X^1 and X^2 be two CTMCs on $\{1, \ldots, n+1\}$ with generators $\mathbf{Q}^1$ and $\mathbf{Q}^2$ and probabilities vectors $\pi^1(t), \pi^2(t)$ at time t.*
Let us assume that $\forall 1 \leq i \leq j \leq n+1$,

$$\forall 1 \leq v < i, \ \sum_{u \leq v} \mathbf{Q}^1[i, u] \geq \sum_{u \leq v} \mathbf{Q}^2[j, u]$$

and

$$\forall j < v \leq n+1, \ \sum_{u \geq v} \mathbf{Q}^1[i, u] \leq \sum_{u \geq v} \mathbf{Q}^2[j, u]$$

Then:

$$\forall v \sum_{u \leq v} \pi^1(0)[u] \geq \sum_{u \leq v} \pi^2(0)[u] \Rightarrow \forall v \, \forall t \ \sum_{u \leq v} \pi^1(t)[u] \geq \sum_{u \leq v} \pi^2(0)[u]$$

The first hypothesis of the theorem states that for any pair of states $i \leq j$, the transition rate in X^1 to go to the set $\{1, \ldots, v\}$ with $v < i$ from state i is bigger than the transition rate in X^2 to go to the same set from state j. Symmetrically, the second hypothesis states that the transition rate in X^1 to go to the set $\{v, \ldots, n+1\}$ with $v > j$ from state i is smaller than the transition rate in X^2 to go the same set from state j. Roughly speaking, X^1 is more likely to go backwards and X^2 is more attracted to go forwards. Thus the conclusion states that given an initial distribution of X^1 more concentrated on the subsets of states of small indices than X^2, in any time in the future the distribution of X^1 will still be more concentrated on such subsets.

Let us suppose that we are given a chain Y with a generator $\widehat{\mathbf{Q}}^{um}$ and a probability vector $\boldsymbol{\pi}_i^Y(0) = \widehat{\boldsymbol{\pi}}_i^a(0) = \mathbf{1}_i$. If Y and $\widehat{X}^a$ fulfill the conditions of the theorem, by choosing $v = n$ in the condition we obtain:

$$\int_0^{\infty} \sum_{j \leq n} \pi_i^Y(t)[j] \, dt \geq \int_0^{\infty} \sum_{j \leq n} \widehat{\pi}_i(t)[j] \, dt = \widehat{h}_i$$

Algorithm 4 - Compute $\mu^{(A)-}$ (Continuous time case)

Input: $\widehat{\mathbf{P}}^{-}$, $\widehat{\mathbf{Q}}^{-}$(upper triangular): $n\times(n+1)$ matrices, $\widehat{\mathbf{Q}}^{+}$: lower triangular $n\times n$ matrix
begin
  compute $\widehat{\mathbf{Q}}^{um}$ from $\widehat{\mathbf{Q}}^{-}$ and $\widehat{\mathbf{Q}}^{+}$ and check the non singularity with algorithm 5
  if $\widehat{\mathbf{Q}}^{um}_{|I}$ is singular **then** halt
  compute $\widehat{\mathbf{h}}^{+} = -\left(\widehat{\mathbf{Q}}^{um}_{|I}\right)^{-1}\cdot \mathbf{1}^T_n$
  compute $\mathbf{V}^{-} = \left(\sum_{k\geq 0}(\widehat{\mathbf{P}}^{-}_{|I})^k\right)$
  $\mu^{(A)-} \leftarrow +\infty$
  for $i_0 \leftarrow 1$ **to** n **do** $\mu^{(A)-} \leftarrow \min(\mu^{(A)-}, \frac{\sum_{j\in I}\mathbf{V}^{-}[i,j]\cdot\widehat{\mathbf{P}}^{-}[j,n+1]}{\widehat{h}^{+}_i})$ **endfor**
end

So that we may choose the following upper bound: $\widehat{h}^{+}_i \stackrel{def}{=} \int_0^\infty \sum_{j\leq n}\pi^Y_i(t)[j]\,dt$. We will detail in the next sub-section an algorithm computing $\widehat{\mathbf{Q}}^{um}$ from the bounding matrices $\widehat{\mathbf{Q}}^{-}$ and $\widehat{\mathbf{Q}}^{+}$. The holding times $\widehat{h}^{+}_i$ in the absorbing CTMC Y, satisfy the following system of linear equations:

$$\forall i\ \in I\ \widehat{h}^{+}_i = \frac{1}{-\widehat{\mathbf{Q}}^{um}[i,i]} + \sum_{j\neq i, j\in I}\frac{\widehat{\mathbf{Q}}^{um}[i,j]}{-\widehat{\mathbf{Q}}^{um}[i,i]}\widehat{h}^{+}_j \Leftrightarrow \forall i\in I\ \sum_{j\in I}\widehat{\mathbf{Q}}^{um}[i,j]\cdot\widehat{h}^{+}_j = -1$$

Transforming the last expression in vectorial notations we get:

$$-\widehat{\mathbf{Q}}^{um}_{|I}\cdot\widehat{\mathbf{h}}^{+} = \mathbf{1}^T_n \Leftrightarrow \widehat{\mathbf{h}}^{+} = -\left(\widehat{\mathbf{Q}}^{um}_{|I}\right)^{-1}\cdot\mathbf{1}^T_n \tag{7}$$

where $\widehat{\mathbf{Q}}^{um}_{|I}$ is the restriction of $\widehat{\mathbf{Q}}^{um}$ to I. Thus the method is applicable if the matrix $\widehat{\mathbf{Q}}^{um}_{|I}$ is non singular. Again we defer the details of such a check until the next sub-section.

4.2 Computing $\widehat{\mathbf{Q}}^{um}$

4.2.1 Presentation of the algorithm

We restate the conditions of theorem 4.1 in terms of matrices $\widehat{\mathbf{Q}}^{a}$ and $\widehat{\mathbf{Q}}^{um}$ making a little transformation in the presentation which gives the key idea of the construction:

$$\forall i,\ \forall 1\leq v<i,\ \sum_{u\leq v}\widehat{\mathbf{Q}}^{um}[i,u] \geq \max_{i\leq j\leq n+1}\left\{\sum_{u\leq v}\widehat{\mathbf{Q}}^{a}[j,u]\right\}$$

and

$$\forall i,\ \forall i<v\leq n+1,\ \sum_{u\geq v}\widehat{\mathbf{Q}}^{um}[i,u] \leq \min_{i\leq j<v}\left\{\sum_{u\geq v}\widehat{\mathbf{Q}}^{a}[j,u]\right\}$$

With this set of inequalities, we build $\widehat{\mathbf{Q}}^{um}$ row per row in decreasing order. Row $n+1$ is irrelevant since $\forall j\ \widehat{\mathbf{Q}}^{a}[n+1,j]=0$. Thus we can set $\forall j\ \widehat{\mathbf{Q}}^{um}[n+1,j]=0$. Since are only interested by $\widehat{\mathbf{Q}}^{um}_{|I}$ the building of this row will be skipped in the algorithm. The remaining rows of $\widehat{\mathbf{Q}}^{a}$ are identical to those of $\widehat{\mathbf{Q}}$. So we will use the bounding matrices in the construction of $\widehat{\mathbf{Q}}^{um}$.

Now looking for the building of row n and examining the above inequalities it appears that only row n of $\widehat{\mathbf{Q}}$ is relevant. We must upper bound the partial sums "from the left" until the diagonal term (excluded) and lower bound the partial sums "from the right" until the diagonal term (excluded). This

Algorithm 5 - Build $\widehat{\mathbf{Q}}^{um}$

Input: $\widehat{\mathbf{Q}}^{-}$(upper triangular): $n \times (n+1)$ matrix, $\widehat{\mathbf{Q}}^{+}$: lower triangular $n \times n$ matrix
begin
  for $j \leftarrow 1$ **to** $n-1$ **do** $\widehat{\mathbf{Q}}^{um}[n,j] \leftarrow \widehat{\mathbf{Q}}^{+}[n,j]$ **endfor**
  $\widehat{\mathbf{Q}}^{um}[n,n+1] \leftarrow \widehat{\mathbf{Q}}^{-}[n,n+1]$
  $\widehat{\mathbf{Q}}^{um}[n,n] \leftarrow -\sum_{j\neq n} \widehat{\mathbf{Q}}^{um}[n,j]$
  for $i \leftarrow n-1$ **downto** 1 **do**
    $x \leftarrow 0; y \leftarrow 0; \Delta \leftarrow 0$ //lower triangle
    for $j \leftarrow 1$ **to** $i-1$ **do**
      $\widehat{\mathbf{Q}}^{um}[i,j] \leftarrow \max(\widehat{\mathbf{Q}}^{+}[i,j] - \Delta, (y-x) + \widehat{\mathbf{Q}}^{um}[i+1,j])$
      $\Delta \leftarrow \Delta + (\widehat{\mathbf{Q}}^{um}[i,j] - \widehat{\mathbf{Q}}^{+}[i,j])$
      $x \leftarrow x + \widehat{\mathbf{Q}}^{um}[i,j]$
      $y \leftarrow y + \widehat{\mathbf{Q}}^{um}[i+1,j]$
    endfor
    $x \leftarrow 0; y \leftarrow 0; \Delta \leftarrow 0$ //upper triangle
    for $j \leftarrow n+1$ **downto** $i+2$ **do**
      $\widehat{\mathbf{Q}}^{um}[i,j] \leftarrow \min(\widehat{\mathbf{Q}}^{-}[i,j] + \Delta, (y-x) + \widehat{\mathbf{Q}}^{um}[i+1,j])$
      $\Delta \leftarrow \Delta + (\widehat{\mathbf{Q}}^{-}[i,j] - \widehat{\mathbf{Q}}^{um}[i,j])$
      $x \leftarrow x + \widehat{\mathbf{Q}}^{um}[i,j]$
      $y \leftarrow y + \widehat{\mathbf{Q}}^{um}[i+1,j]$
    endfor
    $\widehat{\mathbf{Q}}^{um}[i,i+1] \leftarrow \widehat{\mathbf{Q}}^{-}[i,i+1] + \Delta$
    (Ih) **if** $x + \widehat{\mathbf{Q}}^{um}[i,i+1] = 0$ **then** halt
    $\widehat{\mathbf{Q}}^{um}[i,i] \leftarrow -\sum_{j\neq i} \widehat{\mathbf{Q}}^{um}[i,j]$
  endfor
end

gives directly:

$$\forall j < n\ \widehat{\mathbf{Q}}^{um}[n,j] = \widehat{\mathbf{Q}}^{+}[n,j] \text{ and } \widehat{\mathbf{Q}}^{um}[n,n+1] = \widehat{\mathbf{Q}}^{-}[n,n+1]$$

Let us suppose that we have built rows $i+1,\ldots,n$ of $\widehat{\mathbf{Q}}^{um}$. Then for row i, the previous inequalities lead straightforwardly to the definitions:

$$\forall 1 \leq v < i,\ \sum_{u\leq v} \widehat{\mathbf{Q}}^{um}[i,u] = \max(\sum_{u\leq v} \widehat{\mathbf{Q}}^{+}[i,u], \sum_{u\leq v} \widehat{\mathbf{Q}}^{um}[i+1,u])$$

and

$$\forall i+1 < v \leq n+1,\ \sum_{u\geq v} \widehat{\mathbf{Q}}^{um}[i,u] = \min(\sum_{u\geq v} \widehat{\mathbf{Q}}^{-}[i,u], \sum_{u\geq v} \widehat{\mathbf{Q}}^{um}[i+1,u])$$

and

$$\sum_{u\geq i+1} \widehat{\mathbf{Q}}^{um}[i,u] = \sum_{u\geq i+1} \widehat{\mathbf{Q}}^{-}[i,u]$$

Starting from these expressions and "differentiating" them one obtains the algorithm 5 which also includes checking the non singularity of $\widehat{\mathbf{Q}}^{um}_{|I}$.

4.2.2 Checking and characterizing the non singularity of $\widehat{\mathbf{Q}}^{um}_{|I}$

The overall algorithm includes the inversion of $\widehat{\mathbf{Q}}^{um}_{|I}$. A necessary and sufficient condition for this matrix to be invertible, is that the absorbing state $n+1$ must be reachable from any other state [7]. Thus one checks in linear time w.r.t. the number of non null items of the matrix whether it is invertible with the following procedure:

1. Define a oriented graph where the set of nodes is $\{1,\ldots,n+1\}$
2. There is an arc between $i \in I$ and $j \in \{1,\ldots,n+1\}$ iff $\widehat{\mathbf{Q}}^{um}[i,j] \neq 0$
3. Check whether $n+1$ is reachable from any node. This can be done by a breadth-first backward search starting from $n+1$.

The important point here is that the non singularity depends on structural criterium: whether an item is null or not. It would be interesting to have a similar structural characterization depending on $\widehat{\mathbf{Q}}^{+}$ and $\widehat{\mathbf{Q}}^{-}$. This is the goal of next lemma which shows that $\widehat{\mathbf{Q}}^{-}$ is the single significant factor.

Lemma 4.2 *The following assertions are equivalent:*

1. $\widehat{\mathbf{Q}}^{um}_{|I}$ *is invertible*
2. $\forall i \in I\ \sum_{j=i+1}^{n+1} \widehat{\mathbf{Q}}^{-}[i,j] > 0$

Proof.
At first, let us suppose that the assertion 2 is satisfied. Then we know (see the introduction of this section) that: $\forall i \in I\ \sum_{j=i+1}^{n+1} \widehat{\mathbf{Q}}^{um}[i,j] = \sum_{j=i+1}^{n+1} \widehat{\mathbf{Q}}^{-}[i,j] > 0$. Thus in the graph associated to $\widehat{\mathbf{Q}}^{um}$, given $i \in I$ there is an arc from i to some $j > i$. So starting from any node $i \in I$ and following these arcs the node $n+1$ will be eventually reached.

Now suppose that the assertion 2 is not satisfied i.e., $\exists i\ \sum_{j=i+1}^{n+1} \widehat{\mathbf{Q}}^{-}[i,j] = 0$. Thus by definition of $\widehat{\mathbf{Q}}^{um}$, one has $\forall k \leq i,\ \sum_{j=i+1}^{n+1} \widehat{\mathbf{Q}}^{um}[k,j] = 0$, but this means that in the associated graph of the above procedure, the subset of nodes $\{1,\ldots,i\}$ has no outgoing arc. Then $n+1$ is unreachable from this subset of states. $\diamondsuit$

Thus we directly check on the inputs whether our method is applicable. This is done without extra-computation by the instruction (Ih) of algorithm 5 which builds $\widehat{\mathbf{Q}}^{um}$. It should be emphasized that this criterion is really close to the one of the discrete time case.

4.3 Upper bounding the rate

From the method used to compute $\mu^{(A)-}$, we can derive a similar approach to compute an upper bound $\mu^{(A)+}$ of the rate $\mu^{(A)}$. The idea is to lower bound the aggregated probability $\widehat{p}_i^{(B)}$ out of C to B, hence providing an upper bound of $\widehat{p}_i^{(A)}$. We need also to find a lower bound of the holding times $\widehat{h}_i$. Note that this requires only to replace the last column of $\widehat{\mathbf{P}}^{-}$. Without giving technical details, we explain below how to get these lower bounds.

As for the lower bound, we define a matrix $\widehat{\mathbf{Q}}^{lm}$, a monotonic $\leq_{st}$-upper bound of $\widehat{\mathbf{Q}}^{a}$, starting from $\widehat{\mathbf{Q}}^{-}$ and $\widehat{\mathbf{Q}}^{+}$ and $\widehat{\mathbf{P}}^{-}$. Here we suppose to know: $\widehat{\mathbf{Q}}^{+}$ a $n \times (n+1)$ upper-triangular matrix, $\widehat{\mathbf{Q}}^{-}$ a $n \times n$ lower-triangular matrix and $\widehat{\mathbf{P}}^{-}$ a $n \times (n+1)$ lower bounding matrix where the last column gives lower bounds of the probabilities to leave C to B. Despite similar notations, these matrices are not the same ones as matrices of the previous paragraph.

4.3.1 The algorithm

From the conditions of theorem 4.1, we have

$$\forall i,\ \forall 1 \leq v < i,\ \sum_{u \leq v} \widehat{\mathbf{Q}}^{lm}[i,u] \leq \min_{v<j\leq i} \left\{ \sum_{u \leq v} \widehat{\mathbf{Q}}^{a}[j,u] \right\}$$

and

$$\forall i,\ \forall i < v \leq n+1,\ \sum_{u \geq v} \widehat{\mathbf{Q}}^{lm}[i,u] \geq \max_{1\leq j\leq i} \left\{ \sum_{u \geq v} \widehat{\mathbf{Q}}^{a}[j,u] \right\}$$

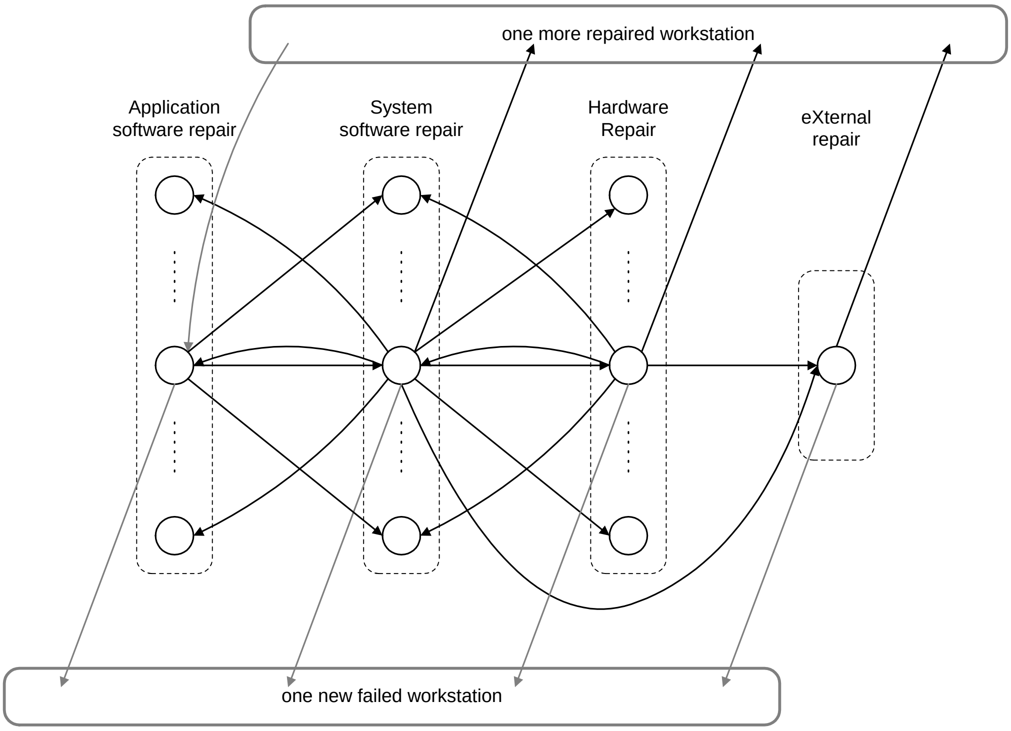

Figure 4: Structure of the repair process

From these inequalities, we build $\widehat{\mathbf{Q}}^{lm}$ row per row in *increasing* order (in contrast with $\widehat{\mathbf{Q}}^{um}$). We start by setting the the first row of $\widehat{\mathbf{Q}}^{lm}$ to the one of $\widehat{\mathbf{Q}}^{+}$. Let us suppose that we have built rows $1,\ldots,i-1$ of $\widehat{\mathbf{Q}}^{lm}$. Then for row i, the previous inequalities lead to the definitions:

$$\forall 1 \leq v < i-1,\ \sum_{u\leq v} \widehat{\mathbf{Q}}^{lm}[i,u] = \min(\sum_{u\leq v} \widehat{\mathbf{Q}}^{-}[i,u], \sum_{u\leq v} \widehat{\mathbf{Q}}^{lm}[i-1,u])$$

and

$$\sum_{u\leq i-1} \widehat{\mathbf{Q}}^{lm}[i,u] = \sum_{u\leq i-1} \widehat{\mathbf{Q}}^{-}[i,u]$$

and

$$\forall i+1 < v \leq n+1,\ \sum_{u\geq v} \widehat{\mathbf{Q}}^{lm}[i,u] = \max(\sum_{u\geq v} \widehat{\mathbf{Q}}^{+}[i,u], \sum_{u\geq v} \widehat{\mathbf{Q}}^{lm}[i-1,u])$$

The algorithm builds the ith row from the $i-1$th row. At first, it sets the terms below the diagonal starting from the first column. Then it computes the terms after the diagonal starting from column $n+1$. Here again $\widehat{\mathbf{Q}}^{lm}_{|I}$ must be invertible. We note that $\widehat{\mathbf{Q}}^{a}_{|I}$ is supposed to be invertible, since in the other case, the output rate from C would be meaningless. Thus, due to the strong order theory, $\widehat{\mathbf{Q}}^{lm}_{|I}$ is also invertible.

5 Example and numerical results for CTMC

We present in this section an example of application of our approach with main numerical results.

5.1 The model

We consider a computing department with several workstations which may fail. On failure, a workstation enters a repair process. Our goal is to find a lower bound of the steady state rate of repair when k workstations are repaired. This corresponds to the rate out of the set C of states of the system with k failed workstations to the set A of states with less than k failed workstations.

k	10	15	20
$\|S\|$	92378	1307504	10015005
$\|S^a\|$	286	816	1771
$\|S\|/\|S^a\|$	323	1602	5655

Table 6: Cardinalities of normal ($|S|$) and aggregated ($|S^a|$) state spaces

The repair process is made of four steps (A,S,H and X) shown in figure 4. In the figure, gray arrows correspond to events not generated by the repair process, and gray boxes correspond to the "environment" of the repair process for a given number of served workstations. Each step U, except the eXternal step, comprises several activities (U_i, with $i = 1,2,3$ in the rest of this paper) with exponential sojourn time (mean $1/q_{U,i}$ and $1/q_X$) and infinite service discipline. For each step, we have only drawn, for clarity, transitions from one (among several) activity.

When entering the repair process, the workstation is first tested for an error at the application level (Application software step). The workstation is then checked for software failure at the operating system level (second step, System software). It reaches activity j with probability p_{A_i,S_j}. After verification of the operating system, the workstation may be repaired (prob. $p_{S_i,R}$), or may need an hardware check (prob. p_{S_i,H_j}), or may be sent to an eXternal company for repair (prob. p_{S_i,A_j}). Moreover, the workstation may also be sent back to the Application step for a new verification (prob. p_{S_i,A_j}). The third step repairs Hardware failures. During this step, the workstation may be repaired (prob. $p_{H_i,R}$), sent to the eXternal repair company (prob. $p_{H_i,X}$) or sent back to the System software step (prob. p_{H_i,S_j}). The external service (usually definitely slower) sends the workstation back to the department after repair.

In order to keep the example simple, we have fixed same branching probabilities out of a step ($p_{U_i,V_j} = p_{U,V}$, $\forall i,j$) for all activities of this step. We also assume that new failures follow a Poisson process with rate q_F. Parameters of the model are then the number m of workstations in the system, the number k of failed workstations, the ten mean sojourn times $1/q_{U,i}, 1/q_X$, the rate q_F and the ten probabilities $p_{U,V}$.

To apply our method, we first need to define the aggregated states. A state of the system is $s = (m-k, i_{H_1}, k_{H_2}, k_{H_3}, k_{S_1}, k_{S_2}, k_{S_3}, k_{A_1}, k_{A_2}, k_{A_3}, k_X)$ where k_{U_i} is the number of workstations served by the ith activity of the step U. Since the behaviours of the workstations are roughly the same in each step, we group together all workstations being repaired by any activity of the Uth step. An aggregated state of the system is then $\widehat{s} = (k_A, k_S, k_H, k_X)$ where $k_U = \sum_i k_{U_i}$ is the number of workstations in step U and $\sum_U k_U = k$, disregarding the behaviour of the up workstations. We choose the order between aggregated states as the reverse lexicographic order: $(k_A,k_S,k_H,k_X) < (k'_A,k'_S,k'_H,k'_X)$ iff $k_A > k'_A$ or $\forall H \leq V < U \leq X$, $k_V = k'_V$ and $k_U > k'_U$. Indeed such an order follows the implicit progression towards a repaired status.

This definition of the aggregated states allows us to deal with large systems. The ratio between the cardinalities of the state space ($|S| = \binom{k+9}{9}$) and the cardinalities of the aggregated state space ($|S^a| = \binom{k+3}{3}$) is given in table 6 for some values of k.

Next, we must define the $\widehat{\mathbf{P}}^-, \widehat{\mathbf{Q}}^-$ and $\widehat{\mathbf{Q}}^+$ matrices. We assume that we only know lower and upper bounding rates q_U^-, q_U^+ out of a stage U and an upper bound q_F^+ of the failure rate q_F.

Although we can define $\widehat{\mathbf{P}}^-$ by $\widehat{\mathbf{P}}^-[i,j] = \frac{\widehat{\mathbf{Q}}^-[i,j]}{\sum_{j\neq i}\widehat{\mathbf{Q}}^+[i,j]}$ this provides a poor lower bound, and moreover it requires to define full matrices $\widehat{\mathbf{Q}}^-$ and $\widehat{\mathbf{Q}}^+$ and not only triangular matrices. It is then always efficient to search for a better lower bounding matrix $\widehat{\mathbf{P}}^-$, depending on the model. In our example, if we move from i to j because a workstation moves from a step U to a step V, we have $\widehat{\mathbf{P}}[i,j] = \frac{p_{U,V}.\widehat{q}_U}{\widehat{q}_U + \sum_{W\neq U}\widehat{q}_W + \widehat{q}_F}$ where $\widehat{q}_Z$ is the (unknown) aggregated rate out of Z in i, $\widehat{q}_F$ is the (unknown) failure rate in i and $p_{U,V}$ is the (known from the model) probability for a workstation in step U to move to V (remember that all stages of a step have same transition probabilities). Since the function $f(x) = \frac{a.x}{x+d}$ is increasing for $a.d > 0$, we can set

$$\widehat{\mathbf{P}}^-[i,j] = \frac{p_{U,V}.i_U.q_U^-}{i_U.q_U^- + \sum_{W\neq U} i_W.q_W^+ + (m-k).q_F^+}$$

	A_j	S_j	H_j	X	R
A_i	0.0	0.33	0.0	0.0	0.0
S_i	0.03	0.0	0.03	0.0	0.82
H_i	0.0	0.03	0.0	0.1	0.81
X	0.0	0.0	0.0	0.0	1.0

Table 7: Routing probabilities between activities (“standard” case)

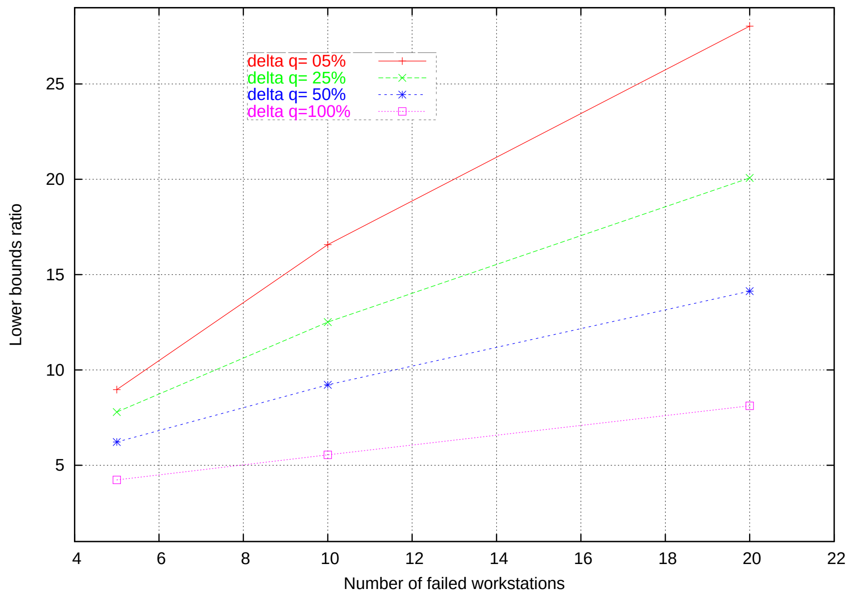

Figure 5: Lower bounds ratios (ordered case, $q^+ = 10^{-3}$)

The definition of $\widehat{\mathbf{Q}}^-$ and $\widehat{\mathbf{Q}}^+$ matrices derives straightforwardly from bounding rates q_U^-, q_U^+ and q_F^+, by examining the possible transitions from (k_A, k_S, k_H, k_X) to (k'_A, k'_S, k'_H, k'_X) taking into account the infinite server discipline.

5.2 Numerical results

We fixed some parameters corresponding to usual situations for all the experiments. The jump probabilities from one activity to another one (identical for all activities of given steps) are given in table 7. Note that there is no repair from the Application step, so that the trivial lower bound $\min_U\{k_U.p_{U,R}.q_U\}$ is 0.

5.2.1 General behaviour

With increasing values of m, it becomes difficult to compute the exact rate μ^A (see table 6). Hence we have compared our results to the method developed by Carrasco [2]. This method is particulary well suited in the context of failure/repair systems especially when failure rates are much smaller (10^{-4}) than repair rates (see Appendix A for a reminder of the Carrasco method). We have computed the bounds with our method and with the Carrasco method for several sets of parameters, $k \leq 20$ and From the model, we assume that the minimal rates out of the activities are $q_A^- = 10.0, q_S^- = 2.0 q_H^- = 1.0, q_X^- = 0.1$ and we did four experiments with $q^+ = 1.05q^-$, $q^+ = 1.25q^-$, $q^+ = 1.5q^-$ and $q^+ = 2.0q^-$. This allows us to test the sensibility of our bound with respect to the initial bounds on the rates. We have also fixed the number of workstations to $m = 100$ and the failure rates ratio $q_f^- = 0.5q_f^+$ in all results reported below. The number k of failed workstations is an important parameter since the service policy is infinite server for each service. We have chosen $k = 5, 10, 20$ for each set of computation.

	A_j	S_j	H_j	X	R
A_i	0.0	0.33	0.0	0.0	0.0
S_i	0.13	0.0	0.13	0.0	0.22
H_i	0.0	0.03	0.0	0.1	0.81
X	0.0	0.0	0.0	0.0	1.0

Table 8: Routing probabilities between activities ("un-ordered case")

Our results are summarized in the figure 5. It gives the ratio between the new bound and the bound obtained with the Carrasco method with rates $q^- = (10.0, 2.0, 1.0, 0.1)$ and standard routing probabilities. We can observe that we always get a better lower bound but depending on the value of k and that this ratio is "under-linear" with k. The quality of our bound is also largely dependent on the information about the rates.

We also computed the previous ratio for a much higher failure rate $q^- = 0.1$ which could be more realistic for systems involving human interactions. With $q^+ = 2.0q^-$ (100% of uncertainty), we get ratios $r = 252, 500, 760$ for $k = 5, 10, 20$. This indicate clearly that our method is well adapted to these kinds of systems.

Finally, we have found, as we expected, that our method is sensitive to the (strong stochastic) ordering of the aggregates. This order depends on the routing probabilities between steps and on the (relative) rates of the activities. For instance, if we assume that System software repair requires a new Application test much often than in the "standard" case (see the new routing probabilities in table 8) we get the following ratios (for $q_f^- = 10^{-3}$ and 100% of q^+/q^- amplitude): $r = 2.09, 2.09, 2.67$ for $k = 5, 10, 20$ to be compared with the values of figure 5.

6 Conclusion

We have presented two methods to derive bounds on steady-state averaged rewards of subsets of states of a MC directly from bounds on the parameters (transition rates or probabilities) of the model which generates the MC.

The first algorithm applies to discrete-time Markov chains (DTMC) and is adaptable to continuous-time Markov chain (CTMC) via an "equivalent" uniformized discrete-time Markov chain. It relies on two key observations. The first observation is that the relative steady-state distribution of a subset of states depends on the steady-state distribution of entering the subset and the mean ratio of visits of a state before leaving the subset. This relation is such that without knowing the former quantity, one can still obtain a bound with the latter one. The second observation is that this mean ratio can be lower bounded by the ratio between a lower bound of the mean number of the visits to a state and a upper bound of the mean number of the visits to all states. Thus we have proposed computations for these two bounds with the help of stochastic ordering theory. Evaluations of this method on a realistic example have shown the accurateness of our bounds with respect to the partial knowledge of the model.

The second algorithm applies to a CTMC for which the reward is the cumulated rate to another subset of the CTMC. Since this reward can be expressed as a ratio similar to the one of the first method, we apply analogous techniques with the help of stochastic ordering theory for CTMCs. Comparisons of this method with the one presented in [2] have shown that these bounds are significantly better with a manageable extra-cost of computation.

The computed bounds are often used as intermediary measures for global bounds on models fulfilling some structural conditions. So we are currently looking for more general models on which such an analysis could still be achieved.

A The Carrasco method for lower bound computation

We remind the reader with the Carrasco method [2] to compute a lower bound of the rate from C (our set of states with k failed components) to A (the set of states with less than k failed components) and we explain how we apply it to our CTMC example of section 5.

The method applies to repair services with Phase-Type (PH) distributions. This method is fact one step of the more general problem to determine a lower bound of the steady-state distribution of A. Each PH distribution is viewed as an absorbing CTMC, Repair being the absorbing state and the server stages being the transient states. Let us assume that there are L PH distributions each having I_l absorbing states. We denote with $\mathbf{B}_l$ the restriction of the generator of the lth PH distribution to the transient states and $\mathbf{b}_l$ the vector of rates from the transient states to the absorbing state (Repaired). Carrasco showed that we have:

$$\mu^{(A)} \geq \frac{\min_{l\leq L, 1\leq i_l\leq I_l}\{p^R_{l,i_l}\}}{\max_{l\leq L, 1\leq i_l\leq I_l}\{h_{l,i_l}\}} \stackrel{def}{=} \alpha \tag{8}$$

where p^R_{l,i_l} is the probability to leave C to Repair starting from i_l, and h_{l,i_l} is the holding time in the transient states starting from i_l. Moreover, the vectors $\mathbf{p}^R_l = (p^R_{l,i_l})$ and $\mathbf{h}_l = (h_{l,i_l})$ may be computed with the following relations,

$$\mathbf{p}^R_l = -(\mathbf{B}_l - f^+)^{-1}.\mathbf{b}_l \quad \text{and} \quad \mathbf{h}_l = -\mathbf{B}_l^{-1}.\mathbf{1} \tag{9}$$

where f^+ is an upper bound of the failure rate from C.

The Carrasco's method applies to a fixed set of L PH distributions. In our example (see section 5), the repair system can be modelled as *one* PH distribution *for a given set x* of exact parameters with:

$$\mathbf{B}_x = \begin{pmatrix}
* & 0 & 0 & q_{A_1,S_1} & q_{A_1,S_2} & q_{A_1,S_3} & 0 & 0 & 0 & q_{A_1,X} \\
0 & * & 0 & q_{A_2,S_1} & q_{A_2,S_2} & q_{A_2,S_3} & 0 & 0 & 0 & q_{A_2,X} \\
0 & 0 & * & q_{A_3,S_1} & q_{A_3,S_2} & q_{A_3,S_3} & 0 & 0 & 0 & q_{A_3,X} \\
q_{S_1,A_1} & q_{S_1,A_2} & q_{S_1,A_3} & * & 0 & 0 & q_{S_1,H_1} & q_{S_1,H_2} & q_{S_1,H_3} & q_{S_1,X} \\
q_{S_2,A_1} & q_{S_2,A_2} & q_{S_2,A_3} & 0 & * & 0 & q_{S_2,H_1} & q_{S_2,H_2} & q_{S_2,H_3} & q_{S_2,X} \\
q_{S_3,A_1} & q_{S_3,A_2} & q_{S_3,A_3} & 0 & 0 & * & q_{S_3,H_1} & q_{S_3,H_2} & q_{S_3,H_3} & q_{S_3,X} \\
0 & 0 & 0 & q_{H_1,S_1} & q_{H_1,S_2} & q_{H_1,S_3} & * & 0 & 0 & q_{H_1,X} \\
0 & 0 & 0 & q_{H_2,S_1} & q_{H_2,S_2} & q_{H_2,S_3} & 0 & * & 0 & q_{H_2,X} \\
0 & 0 & 0 & q_{H_3,S_1} & q_{H_3,S_2} & q_{H_3,S_3} & 0 & 0 & * & q_{H_3,X} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & *
\end{pmatrix} \quad
\mathbf{b}_x = \begin{pmatrix}
q_{A_1,R} \\ q_{A_2,R} \\ q_{A_3,R} \\ q_{S_1,R} \\ q_{S_2,R} \\ q_{S_3,R} \\ q_{H_1,R} \\ q_{H_2,R} \\ q_{H_3,R} \\ q_{X,R}
\end{pmatrix}$$

Since we have bounds for the coefficients of this PH distribution, our method implicitly deals with a continuum of PH distribution whereas Carrasco's method only handles a finite set of PH distributions. Thus in order to compare the two methods we have chosen to select six "strategies" providing six PH distributions. A strategy consists of setting the rate of the nine servers and the failure rate between the minimal and the maximal fixed values. The various strategies range from all minimal rates to all maximal rates

We observed that the probabilities p^R_{l,i_l} do not vary very much with regard to the strategy in contrast with the holding times h_{l,i_l} which we found much larger for two servers with minimal rate and one server with maximal rate.

References

[1] O. Abu-Amsha and J.M. Vincent. An algorithm to bound functionals of Markov chains with large state space. Rapport MAI 25, IMAG, Grenoble, France, April 1996.

[2] J.A. Carrasco. Bounding steady-state availability models with group repair and phase type repair distributions. *Performance Evaluation*, 35:193–214, 1999.

[3] P. J. Courtois and P. Semal. Bounds for the positive eigenvectors of nonnegative matrices and for their approximations by decomposition. *Journal of ACM*, 31(4):804–825, October 1984.

[4] P. J. Courtois and P. Semal. Computable bounds on conditional steady-state probabilities in large markov chains and queueing models. *IEEE Journal on Selected Areas in Communication (SAC)*, 4(6):926–937, September 1986.

[5] S. Donatelli, S. Haddad, P. Moreaux, and M. Sene. Bounds for rewards of systems with clients/servers interactions. In W. J. Stewart B. Plateau and M. Silva, editors, *Proc. of the third International Workshop on Numerical Solution of Markov Chains*, pages 208–227, Zaragoza, Spain, September 8–10 1999. Prensas Universitarias de Zaragoza.

[6] W. Feller. *An introduction to probability theory and its applications. Volume I.* John Wiley & Sons, 1968. (third edition).

[7] W. Feller. *An introduction to probability theory and its applications. Volume II.* John Wiley & Sons, 1971. (second edition).

[8] J.M. Fourneau, M. Lecoz, and F. Quessette. Algorithms for an irreducible an lumpable strong stochastic bound. In A. N. Langville and W. J. Stewart, editors, *Proc. of the International Workshop on Numerical Solution of Markov Chains*, pages 191–206, Urbana, IL, USA, September 3–5 2003. NCSU, Raleigh, NC, USA.

[9] J.C.S. Lui and R.R. Muntz. Computing bounds on steady state availability of repairable computer systems. *Journal of ACM*, 41(4):676–707, July 1994.

[10] S. Mahévas. *Modèles markoviens de grande taille : calculs de bornes*. Thèse, Université de Rennes I, IFSIC, December 18 1997.

[11] S. Mahévas and S. Rubino. Bound computation of dependability and performance measures. *IEEE Transactions on Computers*, 50(5):399–413, 2001.

[12] R.R. Muntz, E. De Souza E Silva, and A. Goyal. Bounding availability of repairable computer systems. In *Proceedings of 1989 ACM SIGMETRICS and PERFORMANCE'89*, May 1989. also in special issue of IEEE Trans. Computers, vol. 38, no. 12, pp. 1714-1723, Dec. 1989.

[13] C. Roos, T. Terlay, and J.Ph. Vial. *Theory and algorithms for linear optimization*. John Wiley & Sons, 1997.

[14] G. Rubino and B. Sericola. Sojourn times in finite Markov processes. *Journal of Applied Probability*, 27:744–756, July 1989.

[15] W. J. Stewart. *Introduction to the numerical solution of Markov chains*. Princeton University Press, USA, 1994.

[16] D. Stoyan. *Comparison Methods for Queueing and Other Stochastic Models*. John Wiley & Sons, 1984.

[17] M. Trémolière, J.M. Vincent, and B. Plateau. Determination of the optimal upper bound of a Markovan generator. Technical Report 106, LGI–IMAG, Grenoble, France, 1992.