Probability in the Engineering and Informational Sciences, **21**, 2007, 235–260. Printed in the U.S.A.
DOI: 10.1017/S0269964807070155

GENERALIZED CLASS $\mathcal{C}$ MARKOV CHAINS AND COMPUTATION OF CLOSED-FORM BOUNDING DISTRIBUTIONS

MOUAD BEN MAMOUN
PRiSM
Université Versailles St Quentin
78035 Versailles, France
and
Université Mohammed V
Rabat, Maroc

ANA BUŠIĆ
PRiSM
Université Versailles St Quentin
78035 Versailles, France

NIHAL PEKERGIN
PRiSM
Université Versailles St Quentin
78035 Versailles, France
and
Centre Marin Mersenne
Université Paris 1
75013 Paris, France
E-mail: {mobe,abusic,nih}@prism.uvsq.fr

In this article we first give a characterization of a class of probability transition matrices having closed-form solutions for transient distributions and the steady-state distribution. We propose to apply the stochastic comparison approach to construct bounding chains belonging to this class. Therefore, bounding chains can be analyzed efficiently through closed-form solutions in order to provide bounds on the distributions of the considered Markov chain. We present algorithms to construct upper-bounding matrices in the sense of the $\leq_{\mathrm{st}}$ and $\leq_{\mathrm{icx}}$ order.

© 2007 Cambridge University Press 0269-9648/07 $25.00

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

1. INTRODUCTION

Markovian models have been largely used in computer systems performance evaluation as well as in different areas of applied probability to analyze quantitatively the performance measures of interest. Despite considerable works on the numerical analysis of Markov chains, the analysis of large-size models still remains a problem [17]. The state space reduction techniques, such as lumpable or quasilumpable Markov chains, and models having specific numerical solutions, such as product-form and matrix geometric solutions, have been largely studied to overcome the state space explosion problem [17]. It is also usual to provide approximations on the performance measures of interest by ignoring or simplifying the complicating aspects of the underlying models. Bounding methods have received attention among other approximative methods since they provide bounds on the performance measures of interest. These methods differ in concepts on which they are based and in types of obtained bound.

The bounding method proposed by Courtois and Semal [6] and then extended by other authors [12,14] provides bounds on steady-state rewards without the computation of the steady-state distribution. Since the analysis is performed in steady state, the problem can be considered as a linear algebra problem, and by using polyhedra theory, the lower and the upper bounds on the steady-state rewards are computed. Although this method provides quite accurate results for some classes of model, there are some constraints on the structures of the underlying Markov chains for its application.

The Markov reward approach proposed by van Dijk [18] and van Dijk and van der Wal [19] provides comparison results for expected cumulative and steady-state rewards of two Markov chains. This approach has been successfully applied to provide bounds for several queuing network models that do not have product-form solutions. The general theorems and their conditions as well as technical steps are given in the survey article [18]. The comparison between the rewards of the original model and the bounding model is established on an expectation basis without providing a comparison of the underlying distributions. Since a specific reward is considered in this approach, the comparison constraints can be, in some cases, relaxed compared to the comparison on a sample path basis. Although there are general comparison and error bounding theorems, the establishment of technical steps of proofs is model driven.

There are also bounding methods and comparison results based on the stochastic comparison that have been largely applied in different areas of applied probability: economics, reliability, and queuing networks. We state here two self-contained books as central references [13,16]. Several stochastic orders have been proposed and the best known is the strong stochastic order $\leq_{st}$. Different comparison results have been established in this stochastic order sense by coupling constructions, by sample-path comparisons, or by means of the stochastic monotonicity. Among the large literature on $\leq_{st}$ comparison results, let us state here some articles that construct bounding Markov chains to overcome the state space explosion problem.

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

In [15], the state space reduction by taking into account the dynamic of the underlying model to establish bounding models has been discussed. In [8,11], the underlying models have been modified to have bounding models that can be analyzed by means of matrix-geometric methods or by the stochastic complementation method. The comparison results are established by sample-path comparison techniques.

Sufficient conditions for the existence of stochastic comparison of two time-homogeneous Markov chains are given by the stochastic monotonicity and bounding properties of their one-step transition probability matrices [13]. In [1], this idea is used to construct a monotone bounding Markov chain in the sense of the $\leq_{\mathrm{st}}$ order on a reduced-size state space. Furthermore, this method is extended to construct monotone, numerically efficient bounding chains [7]. In the sequel, this approach is called the algorithmic stochastic comparison. For a given Markov chain, we construct a monotone bounding matrix that is endowed with some properties that lead to simpler numerical computations or analytical expressions. This approach is not model driven, so it can be applied for any Markov chain; however, the bounds might be not accurate for some cases.

In [3,4], a particular class of Markov chains having a closed-form solution for the steady-state distribution has been introduced. These Markov chains are called class $\mathcal{C}$ Markov chains and the closed-form solution of their transient distributions has been given in [5]. In this article, we first give the extension of class $\mathcal{C}$ Markov chains to $\mathcal{C}^{\mathcal{G}}$ Markov chains also having closed-form solutions for transient and steady-state distributions. Second, we present algorithms to construct $\mathcal{C}^{\mathcal{G}}$ bounding matrices in the sense of the $\leq_{\mathrm{st}}$ and $\leq_{\mathrm{icx}}$ stochastic orders. Therefore, we have the stochastic comparison between the transient and the steady-state distribution of the underlying model and the closed-form bounding distributions.

The variability orders ($\leq_{\mathrm{icx}}$ and $\leq_{\mathrm{icv}}$) are especially used to compare models to emphasize the impact of model parameters or policies, but not to construct bounding Markov chains. Contrary to the $\leq_{\mathrm{st}}$ order, these stochastic orders represent some difficulties in constructing monotone bounding matrices [1]. In [4], an algorithm to construct an $\leq_{\mathrm{icx}}$-monotone, class $\mathcal{C}$ bounding matrix for a given matrix is given. To our best knowledge, this was the first work to construct $\leq_{\mathrm{icx}}$-monotone bounding matrices. In fact, the construction of $\leq_{\mathrm{icx}}$ bounding chains has been possible by means of the specific monotonicity characterization of class $\mathcal{C}$ Markov chains. The generalization of class $\mathcal{C}$ Markov chains will extend the chains that have closed-form solutions to compute transient and the steady-state distributions. Therefore, it would be possible to compute more accurate bounds by constructing monotone bounding chains that belong to this extended class.

The article is organized as follows. In Section 2 we give some preliminaries on the stochastic comparison approach. Section 3 is devoted to class $\mathcal{C}^{\mathcal{G}}$ Markov chains. We first present main properties of this class and the closed-form solutions to compute transient distributions and the steady-state distribution. Then we give algorithms to construct $\leq_{\mathrm{st}}$ and $\leq_{\mathrm{icx}}$ bounding matrices belonging to this class. In Section 4 numerical results are presented in order to give some insights into how

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

the accuracy of bounds might depend on the underlying matrix structure. We conclude in Section 5 and give some perspectives to extend this work.

2. PRELIMINARIES ON THE STOCHASTIC COMPARISON

Here we state the basic definitions and theorems on the stochastic comparison approach. We refer to [13,16] for further details. First, we give the definition to compare two random variables taking values on a totally ordered space $\mathcal{E}$. Let $\mathcal{F}_{\text{st}}$ denote the class of all increasing real functions on $\mathcal{E}$ and $\mathcal{F}_{\text{icx}}$ denote the class of all increasing and convex real functions on $\mathcal{E}$. We denote by $\leq_{\mathcal{F}}$ the stochastic order relation, where $\mathcal{F}$ can be replaced by st or icx to be associated respectively to the class of functions $\mathcal{F}_{\text{st}}$ or $\mathcal{F}_{\text{icx}}$. In the sequel, all of the vectors and matrices are boldfaced, all of the vectors are row vectors, and $\boldsymbol{x}^t$ denotes the transposed vector for $\boldsymbol{x}$. When comparing two vectors or matrices, $\leq$ denotes the componentwise comparison.

DEFINITION 1: *Let X and Y be two random variables taking values on a totally ordered space $\mathcal{E}$,*

$$X \leq_{\mathcal{F}} Y \Leftrightarrow Ef(X) \leq Ef(Y), \quad \forall f \in \mathcal{F},$$

whenever the expectations exist.

Notice that for discrete random variables X and Y with probability vectors $\boldsymbol{p}$ and $\boldsymbol{q}$, the notations $\boldsymbol{p} \leq_{\mathcal{F}} \boldsymbol{q}$ and $X \leq_{\mathcal{F}} Y$ are used interchangeably. Stochastic orders $\leq_{\text{st}}$ and $\leq_{\text{icx}}$ can be also defined by matrices (see [9,10]). Throughout this work, we assume that $\mathcal{E} = \{1, \ldots, n\}$, but the following statements can be extended to the infinite case. We give $\boldsymbol{K}_{\text{st}}$ and $\boldsymbol{K}_{\text{icx}}$ matrices related respectively to the $\leq_{\text{st}}$ and $\leq_{\text{icx}}$ orders. In the sequel, $\boldsymbol{K}_{\mathcal{F}}$ denotes the matrix related to the $\leq_{\mathcal{F}}$ order, $\mathcal{F} \in \{\text{st}, \text{icx}\}$:

$$\boldsymbol{K}_{\text{st}} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 1 & 0 & \ldots & 0 \\ 1 & 1 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \ldots & 1 \end{pmatrix}, \quad \boldsymbol{K}_{\text{icx}} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 2 & 1 & 0 & \ldots & 0 \\ 3 & 2 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n & n-1 & n-2 & \ldots & 1 \end{pmatrix}.$$

PROPOSITION 1: *Let X and Y be two random variables taking values on $\mathcal{E} = \{1, 2, \ldots, n\}$ and let $\boldsymbol{p} = (p_i)_{i=1}^n$ and $\boldsymbol{q} = (q_i)_{i=1}^n$, be probability vectors such that*

$$p_i = \text{Prob}(X = i) \quad \textit{and} \quad q_i = \text{Prob}(Y = i) \quad \textit{for } 1 \leq i \leq n.$$

$$X \leq_{\mathcal{F}} Y \Leftrightarrow \boldsymbol{p}\boldsymbol{K}_{\mathcal{F}} \leq \boldsymbol{q}\boldsymbol{K}_{\mathcal{F}},$$

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

which can be also given as follows:

$$X \leq_{\text{st}} Y \Leftrightarrow \sum_{k=i}^{n} p_k \leq \sum_{k=i}^{n} q_k, \quad \forall i \in \{1, \ldots, n\}, \tag{1}$$

$$X \leq_{\text{icx}} Y \Leftrightarrow \sum_{k=i}^{n} (k-i+1)p_k \leq \sum_{k=i}^{n} (k-i+1)q_k, \quad \forall i \in \{1, \ldots, n\}. \tag{2}$$

It is shown in Theorem 5.2.11 of [13, p. 186] that monotonicity and comparability of the probability transition matrices of time-homogeneous Markov chains yield sufficient conditions to compare the underlying chains stochastically. We first define the monotonicity and comparability of stochastic matrices and then state this theorem.

DEFINITION 2: *Let $\boldsymbol{P}$ be a stochastic matrix. $\boldsymbol{P}$ is said to be stochastically $\leq_{\mathcal{F}}$-monotone if for any probability vectors $\boldsymbol{p}$ and $\boldsymbol{q}$,*

$$\boldsymbol{p} \leq_{\mathcal{F}} \boldsymbol{q} \Rightarrow \boldsymbol{pP} \leq_{\mathcal{F}} \boldsymbol{qP}.$$

DEFINITION 3: *Let $\boldsymbol{P}$ and $\boldsymbol{Q}$ be two stochastic matrices. $\boldsymbol{Q}$ is said to be an upper-bounding matrix of $\boldsymbol{P}$ in the sense of the $\leq_{\mathcal{F}}$ order ($\boldsymbol{P} \leq_{\mathcal{F}} \boldsymbol{Q}$) if*

$$\boldsymbol{PK}_{\mathcal{F}} \leq \boldsymbol{QK}_{\mathcal{F}}.$$

Let us remark that this is equivalent to saying that $\boldsymbol{P} \leq_{\mathcal{F}} \boldsymbol{Q}$ if

$$\boldsymbol{P}_{i,*} \leq_{\mathcal{F}} \boldsymbol{Q}_{i,*}, \quad \forall i \in \{1, \ldots, n\}$$

where $\boldsymbol{P}_{i,*}$ denotes the ith row of matrix $\boldsymbol{P}$.

THEOREM 1: *Let $\boldsymbol{P}$ (resp. $\boldsymbol{Q}$) be the probability transition matrix of the time-homogeneous Markov chain $\{X_n, n \geq 0\}$ (resp. $\{Y_n, n \geq 0\}$). If*

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

then

$$X_n \leq_{\mathcal{F}} Y_n \quad \forall n.$$

The following lemma [13] lets us compare the steady-state distributions of Markov chains, which can be considered as the limiting case ($\Pi_P = \lim_{n\to\infty} X(n)$).

LEMMA 1: *Let $\boldsymbol{Q}$ be a monotone, upper-bounding matrix for a finite stochastic matrix $\boldsymbol{P}$ in the sense of the $\leq_{\mathcal{F}}$ order. If the steady-state distributions ($\boldsymbol{\pi_P}$, $\boldsymbol{\pi_Q}$) exist, then*

$$\boldsymbol{\pi_P} \leq_{\mathcal{F}} \boldsymbol{\pi_Q}.$$

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

3. GENERALIZED CLASS $\mathcal{C}$ ($\mathcal{C}^{\mathcal{G}}$) MARKOV CHAINS

In this section we give the generalization of class $\mathcal{C}$ Markov chains that will be called class $\mathcal{C}^{\mathcal{G}}$ Markov chains. First we present definitions and main properties of this class; then we give algorithms to construct bounding chains.

3.1. Class $\mathcal{C}^{\mathcal{G}}$ Markov Chains and Closed-Form Solutions

DEFINITION 4 (Class $\mathcal{C}^{\mathcal{G}}$): *A stochastic matrix $\boldsymbol{P}$ belongs to $\mathcal{C}^{\mathcal{G}}$ if there are three vectors $\boldsymbol{v}$, $\boldsymbol{c}$, and $\boldsymbol{r}$ in $\mathbb{R}^n$ such that $\boldsymbol{v}$ is a probability vector ($v_i \geq 0$, $\forall i$, $\boldsymbol{v}\boldsymbol{e}^t = \sum_{i=1}^n v_i = 1$), $\boldsymbol{c}\boldsymbol{e}^t = \sum_{i=1}^n c_i = 0$, and*

$$\boldsymbol{P} = \boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c}, \tag{3}$$

where $\boldsymbol{e} = (e_i)_{i=1}^n$ and $e_i = 1$, $\forall i$.

Remark 1: Class $\mathcal{C}$ Markov chains [3] can be obtained by taking $\boldsymbol{r} = (r_i)_{i=1}^n$, $r_i = i - 1$, $\forall i$. In this case, $\boldsymbol{v}$ equals $\boldsymbol{P}_{1,*}$.

In the following lemma we give the properties of power matrices for class $\mathcal{C}^{\mathcal{G}}$ matrices.

LEMMA 2: *Let $\boldsymbol{P}$ be a stochastic matrix from class $\mathcal{C}^{\mathcal{G}}$: $\boldsymbol{P} = \boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c}$. Let α and β be the following constants:*

$$\alpha = \boldsymbol{c}\boldsymbol{r}^t \quad \text{and} \quad \beta = \boldsymbol{v}\boldsymbol{r}^t. \tag{4}$$

The power matrices $\boldsymbol{P}^k, k \geq 2$, belong to class $\mathcal{C}^{\mathcal{G}}$ and they can easily be computed as follows:

$$\boldsymbol{P}^k = \boldsymbol{e}^t\boldsymbol{v} + \left[\alpha^{k-1}\boldsymbol{r}^t + \left(\beta \sum_{i=0}^{k-2} \alpha^i\right)\boldsymbol{e}^t\right]\boldsymbol{c}.$$

PROOF: The proof is done by induction on k.

Basic step ($k = 2$): $\boldsymbol{P}^2 = (\boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c})^2 = \boldsymbol{e}^t\boldsymbol{v}\boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{e}^t\boldsymbol{v}\boldsymbol{r}^t\boldsymbol{c} + \boldsymbol{r}^t\boldsymbol{c}\boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c}\boldsymbol{r}^t\boldsymbol{c}$. Using the fact that $\boldsymbol{v}\boldsymbol{e}^t = 1$ and $\boldsymbol{c}\boldsymbol{e}^t = 0$ (Definition 4), $\boldsymbol{c}\boldsymbol{r}^t = \alpha$ and $\boldsymbol{v}\boldsymbol{r}^t = \beta$ (Eq. (4)), we have: $\boldsymbol{P}^2 = \boldsymbol{e}^t\boldsymbol{v} + [\alpha\boldsymbol{r}^t + \beta\boldsymbol{e}^t]\,\boldsymbol{c}$.

Induction step: Suppose that the lemma holds for k and let us show that it holds also for $k + 1$. Let $\boldsymbol{f} = \alpha^{k-1}\boldsymbol{r} + (\beta\sum_{i=0}^{k-2}\alpha^i)\boldsymbol{e}$. Hence, $\boldsymbol{P}^k = \boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{f}^t\boldsymbol{c}$, so

$$\begin{aligned}
\boldsymbol{P}^{k+1} &= \boldsymbol{P}\boldsymbol{P}^k \\
&= (\boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c})(\boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{f}^t\boldsymbol{c}) \\
&= \boldsymbol{e}^t\boldsymbol{v}\boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{e}^t\boldsymbol{v}\boldsymbol{f}^t\boldsymbol{c} + \boldsymbol{r}^t\boldsymbol{c}\boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c}\boldsymbol{f}^t\boldsymbol{c} \\
&= \boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{e}^t\boldsymbol{v}\boldsymbol{f}^t\boldsymbol{c} + 0 + \boldsymbol{r}^t\boldsymbol{c}\boldsymbol{f}^t\boldsymbol{c}.
\end{aligned}$$

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

On the other hand, $\boldsymbol{v}\boldsymbol{f}^t = \alpha^{k-1}\boldsymbol{v}\boldsymbol{r}^t + (\beta\sum_{i=0}^{k-2}\alpha^i)\boldsymbol{v}\boldsymbol{e}^t = \alpha^{k-1}\beta + (\beta\sum_{i=0}^{k-2}\alpha^i) = \beta\sum_{i=0}^{k-1}\alpha^i$ and $\boldsymbol{c}\boldsymbol{f}^{\mathrm{t}} = \alpha^{k-1}\boldsymbol{c}\boldsymbol{r}^t + (\beta\sum_{i=0}^{k-2}\alpha^i)\boldsymbol{c}\boldsymbol{e}^t = \alpha^k + 0$. Therefore, $\boldsymbol{P}^{k+1} = \boldsymbol{e}^t\boldsymbol{v} + [\alpha^k\boldsymbol{r}^t + (\beta\sum_{i=0}^{k-1}\alpha^i)\boldsymbol{e}^t]\boldsymbol{c}$. ■

We can now state the theorem on the closed-form solution for transient distributions of this class of Markov chains.

THEOREM 2: *Let $\{X(k), k \geq 0\}$ be a time-homogeneous discrete-time Markov chain with probability transition matrix $\boldsymbol{P}$ and let $\boldsymbol{\pi}^k = (\boldsymbol{\pi}_i^k)_{i=1}^n$ be the distribution of $X(k)$.*

If $\boldsymbol{P}$ belongs to class $\mathcal{C}^{\mathcal{G}}$, then for all $k \geq 0$,

$$\boldsymbol{\pi}^k = \boldsymbol{v} + a_k\boldsymbol{c}, \tag{5}$$

where a_k is the constant defined as

$$a_k = \beta\sum_{i=0}^{k-2}\alpha^i + g\alpha^{k-1}$$

$$= \begin{cases} \beta\dfrac{(1-\alpha^{k-1})}{1-\alpha} + g\alpha^{k-1}, & \alpha \neq 1 \\ \beta(k-1) + g, & \alpha = 1. \end{cases}$$

The constants α and β have been already defined in Eq. (4) and g is defined as

$$g = \boldsymbol{\pi}^{\mathbf{0}}\boldsymbol{r}^t.$$

PROOF: Using Lemma 2, we have:

$$\boldsymbol{\pi}^k = \boldsymbol{\pi}^{\mathbf{0}}\boldsymbol{P}^k$$

$$= \boldsymbol{\pi}^{\mathbf{0}}\left(\boldsymbol{e}^t\boldsymbol{v} + \left[\alpha^{k-1}\boldsymbol{r}^t + \left(\beta\sum_{i=0}^{k-2}\alpha^i\right)\boldsymbol{e}^t\right]\boldsymbol{c}\right)$$

$$= \boldsymbol{v} + \alpha^{k-1}g\boldsymbol{c} + \left(\beta\sum_{i=0}^{k-2}\alpha^i\right)\boldsymbol{c}$$

$$= \boldsymbol{v} + a_k\boldsymbol{c}.$$ ■

The following corollary gives the conditions to have the limiting distribution of $\mathcal{C}^{\mathcal{G}}$ matrices.

COROLLARY 1 (Steady-state distribution for class $\mathcal{C}^{\mathcal{G}}$ matrices): *Let $\{X(k), k \geq 0\}$ be a time-homogeneous discrete-time Markov chain with probability transition matrix $\boldsymbol{P}$ from class $\mathcal{C}^{\mathcal{G}}$ and let α be defined as $\alpha = \boldsymbol{c}\boldsymbol{r}^t$. $\{X(k), k \geq 0\}$ has a stationary*

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

distribution if and only if $|\alpha| < 1$. *In that case, the unique stationary distribution (steady-state distribution)* $\boldsymbol{\pi} = \lim_{k\to\infty} \boldsymbol{\pi}^k$ *is given by*

$$\boldsymbol{\pi} = \boldsymbol{v} + R\boldsymbol{c}, \tag{6}$$

where R *is a constant given by* $R = \boldsymbol{v}\boldsymbol{r}^t/(1 - \boldsymbol{c}\boldsymbol{r}^t)$.

PROOF: From Theorem 2 we have $\boldsymbol{\pi}^k = \boldsymbol{v} + a_k\boldsymbol{c}$, $\forall k$, where $\boldsymbol{\pi}^k$ is the distribution of $X(k)$ and $a_k = \beta\sum_{i=0}^{k-2}\alpha^i + g\alpha^{k-1}$. For $|\alpha| < 1$, $\lim_{k\to\infty} a_k = \beta/(1-\alpha)$, so we have $\boldsymbol{\pi} = \boldsymbol{v} + (\beta/(1-\alpha))\boldsymbol{c} = \boldsymbol{v} + R\boldsymbol{c}$, which is independent of $\boldsymbol{\pi}^0$. For $|\alpha| \geq 1$, a_k diverges; thus, the stationary distribution does not exist. ■

Let us now give the conditions to have the limiting distribution in terms of entries of $\boldsymbol{P}$ belonging to class $\mathcal{C}^{\mathcal{G}}$.

Remark 2: Since $\alpha = \boldsymbol{c}\boldsymbol{r}^t = \sum_{i=1}^n c_i r_i = \sum_{i=1}^n (p_{i,i} - v_i) = \sum_{i=1}^n p_{i,i} - 1$, we have $\alpha \geq -1$. The condition $|\alpha| < 1$ is satisfied if and only if $0 < \sum_{i=1}^n p_{i,i} < 2$.

It follows from Theorem 2 and Corollary 1 that the computation of transient distributions and the steady-state distribution are linear with the size of the model.

Remark 3: The complexity of computing $\boldsymbol{\pi}^k$ is $\theta(\max\{n,k\})$ and it equals $\theta(n)$ for $\boldsymbol{\pi}$.

3.2. Monotonicity of Class $\mathcal{C}^{\mathcal{G}}$ Matrices

The following proposition gives sufficient conditions for the monotonicity of class $\mathcal{C}^{\mathcal{G}}$ matrices in terms of vectors $\boldsymbol{c}$ and $\boldsymbol{r}$. For a vector $\boldsymbol{x} \in \mathbb{R}^n$, we will write $\boldsymbol{x} \in \mathcal{F}$ if and only if vector $\boldsymbol{x}$, seen as a real function on $\mathcal{E} = \{1, \ldots, n\}$, is in class $\mathcal{F}$.

PROPOSITION 2: *A matrix* $\boldsymbol{P} = \boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c} \in \mathcal{C}^{\mathcal{G}}$ *is* $\leq_{\mathcal{F}}$*-monotone if*

$$\boldsymbol{c}\boldsymbol{K}_{\mathcal{F}} \geq 0 \quad \textit{and} \quad \boldsymbol{r} \in \mathcal{F}.$$

PROOF: First, we define vectors $\boldsymbol{c}' = \boldsymbol{c}\boldsymbol{K}_{\mathcal{F}}$ and $\boldsymbol{v}' = \boldsymbol{v}\boldsymbol{K}_{\mathcal{F}}$. Hence, $\boldsymbol{P}\boldsymbol{K}_{\mathcal{F}} = \boldsymbol{e}^t\boldsymbol{v}' + \boldsymbol{r}^t\boldsymbol{c}'$. To prove that $\boldsymbol{P}$ is $\leq_{\mathcal{F}}$-monotone, we consider two probability vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ such that $\boldsymbol{p} \leq_{\mathcal{F}} \boldsymbol{q}$. According to Definition 2 we must show that $\boldsymbol{p}\boldsymbol{P} \leq_{\mathcal{F}} \boldsymbol{q}\boldsymbol{P}$, which is equivalent to $\boldsymbol{p}\boldsymbol{P}\boldsymbol{K}_{\mathcal{F}} \leq \boldsymbol{q}\boldsymbol{P}\boldsymbol{K}_{\mathcal{F}}$ (see Proposition 1). We have

$$\begin{aligned}
\boldsymbol{p}\boldsymbol{P}\boldsymbol{K}_{\mathcal{F}} \leq \boldsymbol{q}\boldsymbol{P}\boldsymbol{K}_{\mathcal{F}} &\Leftrightarrow \boldsymbol{p}(\boldsymbol{e}^t\boldsymbol{v}' + \boldsymbol{r}^t\boldsymbol{c}') \leq \boldsymbol{q}(\boldsymbol{e}^t\boldsymbol{v}' + \boldsymbol{r}^t\boldsymbol{c}') \\
&\Leftrightarrow \boldsymbol{v}' + \boldsymbol{p}\boldsymbol{r}^t\boldsymbol{c}' \leq \boldsymbol{v}' + \boldsymbol{q}\boldsymbol{r}^t\boldsymbol{c}' \\
&\Leftrightarrow \boldsymbol{p}\boldsymbol{r}^t\boldsymbol{c}' \leq \boldsymbol{q}\boldsymbol{r}^t\boldsymbol{c}'.
\end{aligned}$$

On the other hand, if X and Y denote random variables corresponding respectively to probability vectors $\boldsymbol{p}$ and $\boldsymbol{q}$, then using Definition 1 and the fact that $\boldsymbol{r} \in \mathcal{F}$, we have $E[r(X)] \leq E[r(Y)]$. Remark that $E[r(X)] = \sum_{k=1}^n r_k p_k = \boldsymbol{p}\boldsymbol{r}^t$ and, similarly,

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

$E[r(Y)] = \boldsymbol{q}\boldsymbol{r}^t$, so $\boldsymbol{p}\boldsymbol{r}^t \leq \boldsymbol{q}\boldsymbol{r}^t$. Using in addition the fact that $\boldsymbol{c}' = \boldsymbol{c}\boldsymbol{K}_{\mathcal{F}} \geq 0$, we conclude that $\boldsymbol{p}\boldsymbol{r}^t\boldsymbol{c}' \leq \boldsymbol{q}\boldsymbol{r}^t\boldsymbol{c}'$, and the proof is achieved. ■

Remark 4: For the $\leq_{\text{st}}$ order, $\boldsymbol{r} \in \mathcal{F}_{\text{st}}$ if and only if vector $\boldsymbol{r}$ is increasing; that is,

$$\boldsymbol{r} \in \mathcal{F}_{\text{st}} \Leftrightarrow r_i \leq r_{i+1}, \quad \forall i, 1 \leq i < n.$$

For the $\leq_{\text{icx}}$ order, $\boldsymbol{r} \in \mathcal{F}_{\text{icx}}$ if and only if vector $\boldsymbol{r}$ is increasing and convex; that is,

$$\boldsymbol{r} \in \mathcal{F}_{\text{icx}} \Leftrightarrow r_1 \leq r_2 \quad \text{and} \quad r_i - r_{i-1} \leq r_{i+1} - r_i, \quad \forall i, 1 < i < n.$$

In the general case, when the underlying matrix $\boldsymbol{P}$ does not belong to class $\mathcal{C}^{\mathcal{G}}$, the monotonicity characterization is given as follows:

- $\boldsymbol{P}$ is $\leq_{\text{st}}$-monotone if and only if for each column j, the function $\sum_{k=j}^{n} P_{i,k}$ is increasing with respect to i [9].
- $\boldsymbol{P}$ is $\leq_{\text{icx}}$-monotone if and only if for each column j, the function $\sum_{k=j}^{n}(k - j + 1)P_{i,k}$ is increasing and convex with respect to i (see [10] for the infinite state space case and [2] for the finite state space case).

3.3. Bounding Algorithms

Obviously, having closed-form solutions makes $\mathcal{C}^{\mathcal{G}}$ matrices attractive for computational issues. However, for a given matrix, the constraints to belong to $\mathcal{C}^{\mathcal{G}}$ matrices might not be satisfied. We propose to construct class $\mathcal{C}^{\mathcal{G}}$ bounding matrices by applying the stochastic comparison approach. In fact, we apply Theorem 1: For a given $\boldsymbol{P}$, we construct $\boldsymbol{Q} \in \mathcal{C}^{\mathcal{G}}$ such that $\boldsymbol{P} \leq_{\mathcal{F}} \boldsymbol{Q}$ and $\boldsymbol{Q}$ is $\leq_{\mathcal{F}}$-monotone. Therefore, we can compute the transient distributions and the steady-state distribution of $\boldsymbol{Q}$ through its closed-form solutions to provide bounding distributions on $\boldsymbol{P}$.

3.3.1. Strong stochastic bounds. In this subsection we give the algorithm to construct bounding matrix in the sense of the $\leq_{\text{st}}$ order. As stated earlier, a $\mathcal{C}^{\mathcal{G}}$ stochastic matrix is defined through three vectors: $\boldsymbol{v}$, $\boldsymbol{r}$, and $\boldsymbol{c}$. For a given matrix $\boldsymbol{P}$ and a nonnegative, increasing vector $\boldsymbol{r}$, Algorithm 1 returns vectors $\boldsymbol{c}$ and $\boldsymbol{v}$ such that $\boldsymbol{Q} = \boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c} \in \mathcal{C}^{\mathcal{G}}$; $\boldsymbol{P} \leq_{\text{st}} \boldsymbol{Q}$ and $\boldsymbol{Q}$ is $\leq_{\text{st}}$-monotone. The nonnegativity constraint is used only to simplify the algorithm and its proof. Some possible choices for $\boldsymbol{r}$ will be discussed in Section 3.4.

Algorithm 1: $\leq_{\text{st}}$-Monotone $\mathcal{C}^{\mathcal{G}}$ Upper Bound $\boldsymbol{Q}$ of $\boldsymbol{P}$

Input: $\boldsymbol{P}$: stochastic matrix; $\boldsymbol{r}$: nonnegative, $\boldsymbol{r} \in \mathcal{F}_{\text{st}}$ (increasing)

Output: $\boldsymbol{Q}$: $\leq_{\text{st}}$-monotone, $\mathcal{C}^{\mathcal{G}}$ upper bound of $\boldsymbol{P}$, $\boldsymbol{Q} = \boldsymbol{e}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c}$

Notation: $K := \min\{i \mid r_i > 0\}, M := \min\{i \mid r_i = r_n\}$

We suppose that $M \neq 1$. (In the other case, the algorithm is trivial: Taking the maximal value of each column of $\boldsymbol{P}\boldsymbol{K}_{\text{st}}$, we obtain a vector $\boldsymbol{x}$. We can then take $\boldsymbol{v} = \boldsymbol{x}\boldsymbol{K}_{\text{st}}^{-1}$; $\boldsymbol{Q} = \boldsymbol{e}^t\boldsymbol{v}$.)

For $M \neq 1$, K is well defined and we have $K \leq M$.

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

//Column n:

1. $v_n = \left[\max_{1 \le i < M} \dfrac{r_n p_{i,n} - r_i}{r_n - r_i} \right]^+;$
2. $c_n = \left[\max_{K \le i \le n} \dfrac{p_{i,n} - v_n}{r_i} \right]^+;$

for $j = n - 1$ **downto** 2 **do**

3. $v_j = \left[\max_{1 \le i < M} \dfrac{r_n \sum_{k=j}^{n} p_{i,k} - r_i}{r_n - r_i} - \sum_{k=j+1}^{n} v_k \right]^+;$
4. $c_j = \max\left(-\dfrac{v_j}{r_n}, \left[\max_{K \le i \le n} \dfrac{\sum_{k=j}^{n} p_{i,k} - \sum_{k=j}^{n} v_k}{r_i} \right]^+ - \sum_{k=j+1}^{n} c_k \right);$

end

//Column 1:

5. $v_1 = 1 - \sum_{k=2}^{n} v_k;$
6. $c_1 = -\sum_{k=2}^{n} c_k;$

We give the following theorem to prove Algorithm 1.

THEOREM 3: *Let $\boldsymbol{P}$ be a stochastic matrix and $\boldsymbol{Q}$ be the matrix computed by Algorithm 1. The matrix $\boldsymbol{Q}$ is a stochastic matrix, $\boldsymbol{Q} \in \mathcal{C}^{\mathcal{G}}$, $\boldsymbol{Q}$ is $\le_{\mathrm{st}}$-monotone and $\boldsymbol{P} \le_{\mathrm{st}} \boldsymbol{Q}$.*

The proof of Theorem 3 is given in the Appendix.

3.3.2. Increasing convex bounds. In Algorithm 2 we give the construction of $\le_{\mathrm{icx}}$ bounding matrices. Let us first define the operators that will be used in this algorithm. $\phi_{\mathrm{icx}} : \mathbb{R}^n \times \mathbb{N} \to \mathbb{R}$ and $\tilde{\phi}_{\mathrm{icx}} : \mathbb{R}^n \times \mathbb{N} \to \mathbb{R}$ are defined as follows:

$$\phi_{\mathrm{icx}}(\boldsymbol{x}, j) = \sum_{k=j}^{n} (k - j + 1) x_k \quad \text{and} \quad \tilde{\phi}_{\mathrm{icx}}(\boldsymbol{x}, j) = \sum_{k=j+1}^{n} (k - j + 1) x_k. \qquad (7)$$

Let us remark that $\phi_{\mathrm{icx}}(\boldsymbol{x}, j) = (\boldsymbol{x}\boldsymbol{K}_{\mathrm{icx}})_j$ and $\phi_{\mathrm{icx}}(\boldsymbol{x}, j) = \tilde{\phi}_{\mathrm{icx}}(\boldsymbol{x}, j) + x_j$.

Algorithm 2: $\le_{\mathrm{icx}}$-Monotone $\mathcal{C}^{\mathcal{G}}$ Upper Bound $\boldsymbol{Q}$ of $\boldsymbol{P}$

Input: $\boldsymbol{P}$: stochastic matrix; $\boldsymbol{r}$: nonnegative vector, $\boldsymbol{r} \in \mathcal{F}_{\mathrm{icx}}$ (increasing, convex)

Output: $\boldsymbol{Q}$: $\le_{\mathrm{icx}}$-monotone $\mathcal{C}^{\mathcal{G}}$ upper bound of $\boldsymbol{P}$, $\boldsymbol{Q} = \boldsymbol{e}^t \boldsymbol{v} + \boldsymbol{r}^t \boldsymbol{c}$

Notation: $K := \min\{i \mid r_i > 0\}, M := \min\{i \mid r_i = r_n\}$

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

We suppose that $M \neq 1$. (In the other case, the algorithm is trivial: Taking the maximal value of each column of $\boldsymbol{PK}_{\text{icx}}$, we obtain a vector $\boldsymbol{x}$. We can then take $\boldsymbol{v} = \boldsymbol{x}\boldsymbol{K}_{\text{icx}}^{-1}$; $\boldsymbol{Q} = \boldsymbol{e}^t\boldsymbol{v}$.)

For $M \neq 1$, K is well defined and we have $K \leq M$.

//Column n:

1. $v_n = \left[\max_{1\leq i<M} \dfrac{r_n p_{i,n} - r_i}{r_n - r_i}\right]^+$;
2. $c_n = \left[\max_{K\leq i\leq n} \dfrac{p_{i,n} - v_n}{r_i}\right]^+$;

for: $j = n - 1$ **downto** 2 **do**

3. $v_j = \left[\max_{1\leq i<M} \dfrac{r_n(\phi_{\text{icx}}(P_{i,*}, j) - \phi_{\text{icx}}(\boldsymbol{v}, j+1) - r_i\phi_{\text{icx}}(\boldsymbol{c}, j+1)) - r_i}{r_n - r_i} - \sum_{k=j+1}^{n} v_k\right]^+$;
4. $c_j = \max\left(-\dfrac{v_j}{r_n}, \left[\max_{K\leq i\leq n} \dfrac{\phi_{\text{icx}}(P_{i,*}, j) - \phi_{\text{icx}}(\boldsymbol{v}, j)}{r_i}\right]^+ - \tilde{\phi}_{\text{icx}}(\boldsymbol{c}, j)\right)$;

end

//Column 1:

5. $v_1 = 1 - \sum_{k=2}^{n} v_k$;
6. $c_1 = -\sum_{k=2}^{n} v_k$;

The following theorem is given to prove Algorithm 2.

THEOREM 4: *Let $\boldsymbol{P}$ be a stochastic matrix and $\boldsymbol{Q}$ be the matrix computed by Algorithm 2. The matrix $\boldsymbol{Q}$ is a stochastic matrix from class $\mathcal{C}^{\mathcal{G}}$ and it is an $\leq_{\text{icx}}$-monotone upper bound for $\boldsymbol{P}$.*

The proof of Theorem 4 is given in the Appendix.

Remark 5: The computation complexity of Algorithm 1 and Algorithm 2 is $\theta(n^2)$.

3.4. Choosing Vector *r*

For both algorithms, vector $\boldsymbol{r}$ of row increments, satisfying monotonicity conditions (see Proposition 2), is fixed. Moreover, vector $\boldsymbol{r}$ is also supposed to be nonnegative in order to simplify the algorithm and its proof. Thus, any vector satisfying the above conditions can be chosen as the vector of row increments. We present some of possible choices for vector $\boldsymbol{r}$ in the following.

If we take vector $\boldsymbol{r} = (r_i)_{i=1}^n$, $r_i = i - 1$, we obtain the bounding matrix that belongs to class $\mathcal{C}$ (see Remark 1). It is important to notice that this choice is independent of the underlying matrix entries.

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

We propose some other possibilities for vector $\boldsymbol{r}$ in order to compute more accurate bounds. The main idea here is to improve bounding entries in the last column. In fact, since in both algorithms the construction is done recursively from the last column to the first column, the perturbations in the last column entries of the bounding matrix are propagated to other bounding matrix entries. So by diminishing the perturbations in the last column, it might be possible to improve the accuracy of the bounds. The other reason comes essentially from performance evaluation studies. Quite often, we are interested in bounding only the tail distribution. For instance, in reliability applications we are interested in bounding the probability that the studied system is not operational; in queuing applications we bound the overflow probability of the buffer. In general, these measures concern a few states and they can be ordered as last states. Thus, accurate bounds for last states would be sufficient for these cases.

3.4.1. Strong stochastic case. For the strong stochastic order, there is a vector $\boldsymbol{r}$ that minimizes the differences between the exact and the bounding entries for the last column.

PROPOSITION 3: *Let the input vector $\boldsymbol{r}$ of Algorithm 1 be defined as*

$$r_i = \max_{k \le i} p_{k,n}, \quad \forall i. \tag{8}$$

Then the bounding matrix $\boldsymbol{Q}$ computed by Algorithm 1 satisfies the following:

$$\forall \boldsymbol{U} \text{ such that } \boldsymbol{P} \le_{\text{st}} \boldsymbol{U} \text{ and } \boldsymbol{U} \text{ is } \le_{\text{st}}\text{-monotone}, \quad |q_{i,n} - p_{i,n}| \le |u_{i,n} - p_{i,n}|, \quad \forall i. \tag{9}$$

PROOF: $\boldsymbol{P} \le_{\text{st}} \boldsymbol{U}$ and $\boldsymbol{P} \le_{\text{st}} \boldsymbol{Q}$ (see Theorem 3) yield that $p_{i,n} \le u_{i,n}$ and $p_{i,n} \le q_{i,n}$, $\forall i$, so Eq. (9) is equivalent to $q_{i,n} \le u_{i,n}$, $\forall i$. Since $\boldsymbol{U}$ is $\le_{\text{st}}$-monotone, $u_{i,n}$ are increasing in i. Moreover, $u_{i,n} \ge p_{i,n}$, $\forall i$, so $u_{i,n} \ge \max_{k \le i} p_{k,n} = r_i$, $\forall i$.

Finally, we will show that Algorithm 1 with the input vector $\boldsymbol{r}$ defined by Eq. (9) yields $v_n = 0$ and $c_n = 1$. Therefore, $q_{i,n} = v_n + r_i c_n = r_i \le u_{i,n}$, $\forall i$, and the proof is concluded. Indeed, we have $r_n - r_i > 0$, $1 \le i < M$, $r_i = \max_{k \le i} p_{k,i} \ge p_{i,n}$, $\forall i$, and $r_n \le 1$; thus, it can be seen from the first line of Algorithm 1 that $v_n = 0$. On the other hand, $p_{i,n} \le r_i$, $\forall i$, and $r_K = p_{K,n}$ so it follows from the second line of Algorithm 1 that $c_n = 1$. ■

In the sequel, vector $\boldsymbol{r}$ obtained by Eq. (8) will be called the St-LC vector. In the following example, we present class $\mathcal{C}^{\mathcal{G}}$ matrices obtained with vector $\boldsymbol{r}$ corresponding to class $\mathcal{C}$ and with the St-LC vector computed from Eq. (8).

Example 1: Let $\boldsymbol{P}$ be the original matrix with the steady-state distribution $\boldsymbol{\pi_P} = (\frac{5}{17}, \frac{6}{17}, \frac{6}{17}) \approx (0.29412, 0.35294, 0.35294)$. By taking vector $\boldsymbol{r}$ that corresponds to class $\mathcal{C}$, $\boldsymbol{r} = (0, 1, 2)$, Algorithm 1 returns matrix $\boldsymbol{A}$ with vector $\boldsymbol{v} = (0.4, 0.6, 0)$ and vector $\boldsymbol{c} = (-0.2, -0.2, 0.4)$. The steady-state distribution of $\boldsymbol{A}$ is $\boldsymbol{\pi_A} = (0.1, 0.3, 0.6)$.

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

The St-LC vector computed from Eq. (8) is $\boldsymbol{r} = (0, 0.4, 0.6)$. In this case, Algorithm 1 returns bounding matrix $\boldsymbol{A}'$ with vector $\boldsymbol{v} = (0, 0.4, 0.6)$ and vector $\boldsymbol{c} = (-0.5, -0.5, 1)$. The steady-state distribution of this bounding matrix is $\boldsymbol{\pi}_{\boldsymbol{A}'} = (0.2, 0.4, 0.4)$.

$$\boldsymbol{P} = \begin{pmatrix} 0.4 & 0.6 & 0 \\ 0.2 & 0.4 & 0.4 \\ 0.3 & 0.1 & 0.6 \end{pmatrix}, \qquad \boldsymbol{A} = \begin{pmatrix} 0.4 & 0.6 & 0 \\ 0.2 & 0.4 & 0.4 \\ 0 & 0.2 & 0.8 \end{pmatrix},$$

$$\boldsymbol{A}' = \begin{pmatrix} 0.4 & 0.6 & 0 \\ 0.2 & 0.4 & 0.4 \\ 0.1 & 0.3 & 0.6 \end{pmatrix}.$$

We note that $\boldsymbol{A}'$ belongs to class $\mathcal{C}^{\mathcal{G}}$ but not to class $\mathcal{C}$ and $\boldsymbol{\pi}_{\boldsymbol{P}} \leq_{\text{st}} \boldsymbol{\pi}_{\boldsymbol{A}'} \leq_{\text{st}} \boldsymbol{\pi}_{\boldsymbol{A}}$. However, we do not always have this relation (see Fig. 4).

3.4.2. Increasing convex case. In the case of the increasing convex order, for a given matrix $\boldsymbol{P}$ it is not always possible to find a monotone upper-bounding matrix $\boldsymbol{Q}$ satisfying the following conditions:

$$\forall\ \boldsymbol{U} \text{ such that } \boldsymbol{P} \leq_{\text{icx}} \boldsymbol{U} \text{ and } \boldsymbol{U} \text{ is } \leq_{\text{icx}}\text{-monotone}, \quad |q_{i,n} - p_{i,n}| \leq |u_{i,n} - p_{i,n}|, \quad \forall i. \tag{10}$$

Let us show this for matrix $\boldsymbol{P}$ of Example 1. It can be shown that the following matrices $\boldsymbol{B}$ and $\boldsymbol{B}'$ are $\leq_{\text{icx}}$-monotone upper-bounding matrices for $\boldsymbol{P}$. To satisfy Eq. (10) we must have $p_{i,n} \leq q_{i,n} \leq \min(b_{i,n}, b'_{i,n})$, $\forall i$; thus, $q_{1,3} = 0$, $q_{2,3} = 0.4$, and $q_{3,3} = 0.6$. However, this vector $\boldsymbol{r} = (0, 0.4, 0.6)$ is not compatible with the monotonicity constraints (see Remark 4).

$$\boldsymbol{B} = \begin{pmatrix} 0.4 & 0.6 & 0 \\ 0.2 & 0.4 & 0.4 \\ 0 & 0.2 & 0.8 \end{pmatrix}, \qquad \boldsymbol{B}' = \begin{pmatrix} 0.4 & 0.4 & 0.2 \\ 0.2 & 0.4 & 0.4 \\ 0 & 0.4 & 0.6 \end{pmatrix}.$$

Let us recall that an $\leq_{\text{icx}}$-monotone upper-bounding matrix $\boldsymbol{Q}$ of $\boldsymbol{P}$ is optimal in the sense of $\leq_{\text{icx}}$ order if

$$\forall\ \boldsymbol{U} \text{ such that } \boldsymbol{P} \leq_{\text{icx}} \boldsymbol{U} \text{ and } \boldsymbol{U} \text{ is } \leq_{\text{icx}}\text{-monotone}, \quad \boldsymbol{P} \leq_{\text{icx}} \boldsymbol{Q} \leq_{\text{icx}} \boldsymbol{U}. \tag{11}$$

In fact, Eq. (10) represents the optimality conditions for the last column, so it is a necessary condition for the global optimality [Eq. (11)]. Hence, the previous example shows that it is not always possible to find the optimal bound in the $\leq_{\text{icx}}$ sense. Moreover, this is the case for any stochastic matrix, not only for class $\mathcal{C}^{\mathcal{G}}$ matrices, since the properties of this class have not been considered.

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

As we cannot always satisfy Eq. (10), we propose to minimize the sum of distances between $q_{i,n}$ and $p_{i,n}$:

$$\sum_{i=1}^{n} |q_{i,n} - p_{i,n}|. \tag{12}$$

PROPOSITION 4: *Let vector* $\boldsymbol{r}$ *be a solution of the following problem: Find vector* $\boldsymbol{x}$ *with a minimal sum of entries satisfying the monotonicity (*$\boldsymbol{x} \in \mathcal{F}_{\text{icx}}$*) and the comparison constraints,*

$$\textit{minimize} \sum_{i=1}^{n} x_i \textit{ satisfying}$$
$$\begin{cases} x_i - p_{i,n} \geq 0, & \forall i \\ -x_1 + x_2 \geq 0 \\ x_i - 2x_{i+1} + x_{i+2} \geq 0, & \forall i \leq n-2 \\ -x_n + 1 \geq 0 \end{cases} \tag{13}$$

Then the upper-bounding matrix $\boldsymbol{Q}$ *computed by Algorithm 2 with this input vector* $\boldsymbol{r}$ *minimizes Eq. (12); that is,*

$$\forall\ \boldsymbol{U} \textit{ such that } \boldsymbol{P} \leq_{\text{icx}} \boldsymbol{U} \textit{ and } \boldsymbol{U} \textit{ is } \leq_{\text{icx}}\textit{-monotone}, \quad \sum_{i=1}^{n} |q_{i,n} - p_{i,n}| \leq \sum_{i=1}^{n} |u_{i,n} - p_{i,n}|. \tag{14}$$

PROOF: Let us remark first that $\boldsymbol{P} \leq_{\text{icx}} \boldsymbol{U}$ and $\boldsymbol{P} \leq_{\text{icx}} \boldsymbol{Q}$ (see Theorem 4) give $p_{i,n} \leq u_{i,n}$ and $p_{i,n} \leq q_{i,n}$, $\forall i$; so Eq. (14) is equivalent to $\sum_{i=1}^{n} q_{i,n} \leq \sum_{i=1}^{n} u_{i,n}$. Constraints (13) are fulfilled by all $\leq_{\text{icx}}$-monotone upper bounds of $\boldsymbol{P}$, so we have $\sum_{i=1}^{n} r_i \leq \sum_{i=1}^{n} u_{i,n}$, $\forall \boldsymbol{U} \leq_{\text{icx}}$-monotone upper bound of $\boldsymbol{P}$.

Similarly to the proof of Proposition 3, it can be shown from Algorithm 2 that $v_n = 0$ and $c_n \leq 1$. Moreover, if $c_n < 1$, then $\sum_{i=1}^{n} q_{i,n} = c_n \sum_{i=1}^{n} r_i < \sum_{i=1}^{n} r_i$, which is in contradiction with the fact that $\boldsymbol{r}$ is a solution of (13). Thus, we have $c_n = 1$. ■

The Simplex algorithm can be used to find a solution of problem (13). Let us remark that solution of this problem is not necessarily unique. The bounds computed from Algorithm 2 with input vector $\boldsymbol{r}$ is obtained through the Simplex algorithm will be called Icx-Simplex bounds.

Although problem (13) can be efficiently solved by the Simplex algorithm, we cannot guarantee its complexity in the worst case. Therefore, we present an alternative heuristic choice for $\boldsymbol{r}$. Unrolling the comparison [Eq. (2)] and the monotonicity (Remark 4) constraints for the last column increasingly in rows,

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

$$
\begin{aligned}
r_1 &= p_{1,n}, \\
r_2 &= \max(p_{2,n}, r_1), \\
r_i &= \max(p_{i,n}, 2r_{i-1} - r_{i-2}), \quad \forall i > 2,
\end{aligned} \tag{15}
$$

we obtain a nonnegative vector $\boldsymbol{r}$ such that $\boldsymbol{r} \in \mathcal{F}_{\text{icx}}$. In the sequel, the bounds obtained from Algorithm 2 with this input vector will be called Icx-LC bounds.

Let us remark that a vector $\boldsymbol{r}$ obtained by relations (15) satisfy constraints (13) except the last one, $r_n \leq 1$. However, this will be compensated by the fact that we will have $c_n < 1$ during the construction of the bounding matrix through Algorithm 2. Thus, the last column of the bounding matrix $\boldsymbol{Q}$ computed by Algorithm 2 satisfies all of the constraints (13). Although the Icx-LC bound does not necessarily minimize Eq. (12), since we take into consideration the information contained in the last column of the original matrix, this choice could yield better bounds than class $\mathcal{C}$ bounds for rewards that are mostly based on the last states. This is true in the case of the example studied in Section 4. Moreover, for this example, Icx-LC bounds are remarkably close to Icx-Simplex bounds.

Remark 6: The complexity of computing different vectors $\boldsymbol{r}$ is not important compared to that of Algorithms 1 and 2 (Remark 5). In fact, Eqs. (8) and (15) can be solved with linear complexity, and Eq. (13) can be solved efficiently with the Simplex algorithm.

4. NUMERICAL EXAMPLE

Let us first emphasize that the proposed bounding method can be applied to any finite discrete-time Markov chain, but the tightness of bounds depends largely on the underlying transition matrices. Our main goal in this section is to give some insights on the relationship between the accuracy of $\mathcal{C}^{\mathcal{G}}$ bounds and the underlying matrix structures. We study an example that has been also studied in previous articles because of its simplicity in generating different structures of probability transition matrices.

We consider a finite-capacity buffer with multiple servers and batch arrivals. The time is supposed to be divided into slots and the analysis is provided in discrete time. The number of servers is denoted by S and the buffer capacity by B; thus, $C = B + S$ is the maximum number of customers in the system (see Fig. 1). The size of batch arrivals is assumed to be distributed according to a truncated modified geometrical distribution with parameter p_a. The probability that i customers arrive during a slot is $a_i = (1 - p_a)p_a^i$, $1 \leq i \leq A$, where A is the maximal batch size. The service time is distributed geometrically, and p_s denotes the probability that the service is finished in one slot. Obviously, the values of S and A have a large impact on the sparseness of the matrix. The number of nonzero entries increases with the values of S and A and the probability transition matrix is full when $A = S = B$. The probabilities p_a and p_s for arrivals and services, respectively, determine where

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

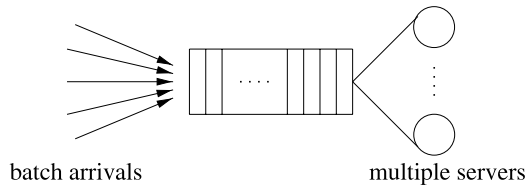


FIGURE 1. Considered system.

the probability mass is located in the matrix. In the following figures the curves of bounds are noted according to vector $\boldsymbol{r}$ used as input in bounding algorithms (see Section 3.4).

In Figure 2, the values of B, p_a, and p_s are fixed and we assume that $S = A$. We vary this last value and give the average number of customers in the steady state. We note that when $S = A$ increases, the accuracy of bounds is improved for all class $\mathcal{C}^{\mathcal{G}}$ bounds for both $\leq_{\text{st}}$ and $\leq_{\text{icx}}$ orders. In fact, whatever the structure of the original matrix, the bounding matrix is filled due to the structure of class $\mathcal{C}^{\mathcal{G}}$ matrices. Thus, if the number of zero entries in the original matrix is large, the number of modified entries would be also large and this might cause a deterioration of the accuracy of bounds. However, we note that it is possible to have tight bounds even for relatively sparse matrices, especially with the $\leq_{\text{icx}}$ order. In fact, the presented results confirm the intuition which follows from the structure of $\mathcal{C}^{\mathcal{G}}$ matrices that the proposed bounds are more adapted for full matrices. On the other hand, it is not possible to state that the bounds for relatively sparse matrices are always loose, since the quality of bounds depends also on the magnitude of the perturbations.

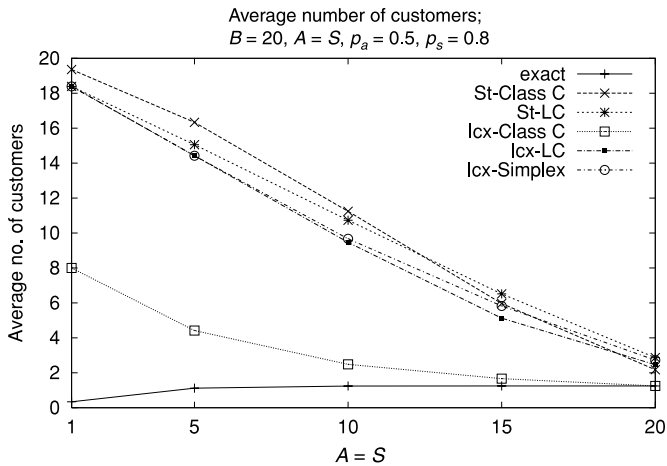


FIGURE 2. Impact of sparseness of the transition matrix on the quality of class $\mathcal{C}^{\mathcal{G}}$ bounds (steady state).

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

In the case of the $\leq_{st}$ order, we can see in Figure 2 that the St-LC and the St-Class C curves cross each other: The class $\mathcal{C}$ bound is tighter for large values of $A = S$ (full matrices) and the St-LC bound is tighter for sparser matrices. Thus, each of these $\leq_{st}$ bounds might be of interest. In the case of the $\leq_{icx}$ order, class $\mathcal{C}$ bounds are tighter than the other considered class $\mathcal{C}^{\mathcal{G}}$ bounds and they are very close to the exact values for full matrices. Moreover, there is not a sensible difference between Icx-LC and Icx-Simplex bounds. This phenomena has been observed not only in Figure 2 but also for different sets of parameters that we considered. Hence, the heuristic proposed in Section 3.4.2 seems to be useful to avoid the complexity of the Simplex algorithm.

We also note in Figure 2 that $\leq_{icx}$ bounds are generally more accurate than the $\leq_{st}$ bounds. This is due to specific monotonicity properties of class $\mathcal{C}^{\mathcal{G}}$ matrices. In fact, the $\leq_{st}$ comparison of random variables and stochastic matrices always implies the $\leq_{icx}$ comparison [13], as the class of increasing convex functions is contained in the class of increasing functions. For the monotonicity, this is not generally the case. However, for class $\mathcal{C}$ matrices, the monotonicity in the $\leq_{st}$ sense implies the monotonicity in the $\leq_{icx}$ sense [4]. We can also show that, for $\mathcal{C}^{\mathcal{G}}$ matrices, if the chosen vector $\boldsymbol{r}$ is increasing and convex ($\boldsymbol{r} \in \mathcal{F}_{icx}$), then $\boldsymbol{cK}_{st} \geq 0$ implies the $\leq_{icx}$ monotonicity (see Proposition 2). On the other hand, $\leq_{st}$ bounds might be useful if one wants to compute bounds on increasing but not convex rewards. For instance, if we want to compare the tail distributions, we cannot use $\leq_{icx}$ bounds.

Let us remember that the proposed approach can be also applied to provide transient bounds. In Figure 3 we present transient behaviors of the average number of customers for the parameter values of Figure 2 with $A = S = 10$. We can see here that in the transient case we can make the same observations as in the steady-state case (Figure 2). More generally, we have noticed also through other experiments

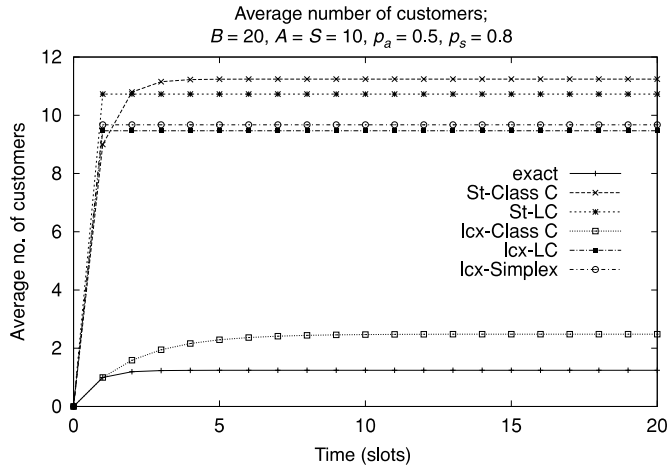


FIGURE 3. Transient $\mathcal{C}^{\mathcal{G}}$ bounds for the case $A = S = 10$.

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

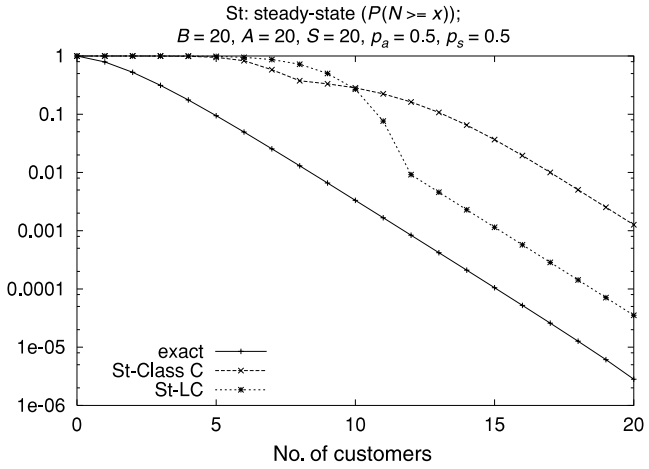


FIGURE 4. Steady-state analysis ($\leq_{\text{st}}$ order).

that the quality of transient bounds has the same tendency as in the case of steady-state bounds.

In Figures 4 and 5, we consider bounds on steady-state distributions. Let N be the number of customers in the steady-state case. In Figure 4, we present $\leq_{\text{st}}$ bounds for the tail distribution ($\text{Prob}(N \geq x)$). For $\leq_{\text{icx}}$ bounds, it is not possible to com-

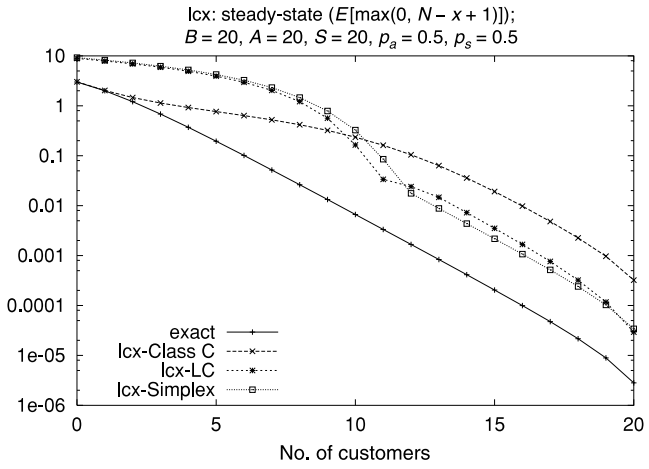


FIGURE 5. Steady-state analysis ($\leq_{\text{icx}}$ order).

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

pare tail distributions. However, we have the following characterization of the $\leq_{\text{icx}}$ order (see [13]):

$$X \leq_{\text{icx}} Y \Leftrightarrow E[(X-x)^+] \leq E[(Y-x)^+], \quad \forall x.$$

Informally speaking, this characterization says that, for a given level x, we compare the mean overflow from that level. In Figure 5, we compare $E[(N-x+1)^+]$ for different $\leq_{\text{icx}}$ bounds. We note that for this set of parameters, the bounds obtained using vectors $\boldsymbol{r}$ derived from the last column are more accurate for both $\leq_{\text{icx}}$ and $\leq_{\text{st}}$ bounds. For instance, in Figures 4 and 5, for $x \geq 10$ we obtain tighter bounds by taking into account the last column of the original matrix.

It is not surprising to obtain more accurate bounds for last states (i.e., bounds on the rewards of last states), by taking into account the information contained in the last column of the original matrix. In fact the information used to derive the input vector $\boldsymbol{r}$ is valid for the last column and also, to a lesser extent, for the columns that are relatively close to it. However, it is less intuitive that we obtain tighter results from class $\mathcal{C} \leq_{\text{icx}}$ bounds when studying average behavior of the underlying model.

It can be seen from Figures 6 and 7 that different class $\mathcal{C}^{\mathcal{G}}$ bounds can cross each other several times. Due to the low computational complexity of class $\mathcal{C}^{\mathcal{G}}$ bounds, it is worth computing different class $\mathcal{C}^{\mathcal{G}}$ bounds and taking the best one for each value. In Table 1 we give the best $\leq_{\text{st}}$ and $\leq_{\text{icx}}$ bounds that can be derived from the bounds of Figures 6 and 7. In Table 2 we present, for the same set of parameters, the best transient bounds for $\text{Prob}(N \geq 15)$ for the $\leq_{\text{st}}$ order and $E[(N-14)^+]$ for the $\leq_{\text{icx}}$ order. We observe that the best transient bounds are obtained from different vectors $\boldsymbol{r}$, as in the steady-state case.

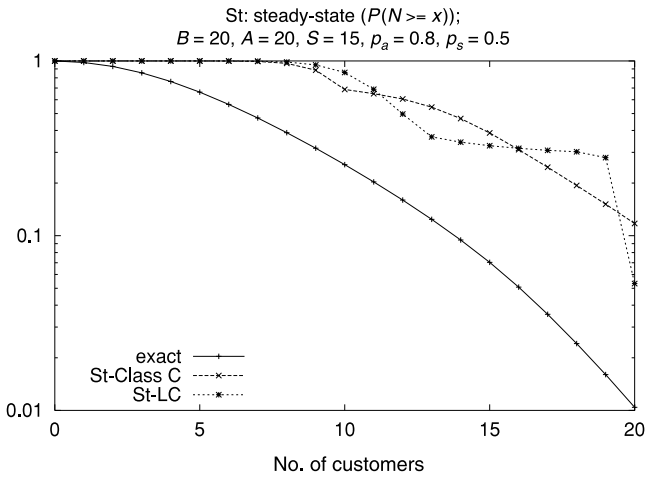


FIGURE 6. Bounds can cross more than once ($\leq_{\text{st}}$ case).

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

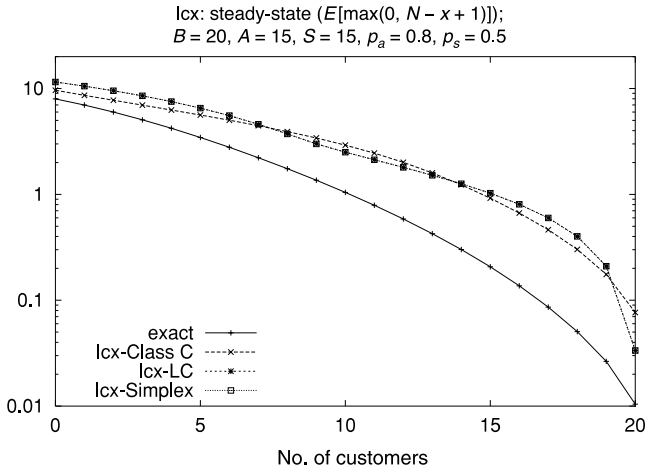


FIGURE 7. Bounds can cross more than once ($\leq_{\text{icx}}$ case).

5. CONCLUSION

In this article we present the generalization of class $\mathcal{C}$ matrices to class $\mathcal{C}^{\mathcal{G}}$ matrices. This class of Markov chains has closed-form solutions to compute transient distributions and the steady-state distribution, which makes it attractive for computational issues of large Markov chains. We propose to apply the algorithmic stochastic comparison approach to construct upper-bounding matrices belonging to this class. For a given finite Markov chain we construct a bounding class $\mathcal{C}^{\mathcal{G}}$ chain and compute its distributions to provide bounding distributions. By analyzing bounding class $\mathcal{C}^{\mathcal{G}}$ chains, we significantly gain in terms of both computational and storage complexity, compared to the exact analysis.

We give algorithms to construct class $\mathcal{C}^{\mathcal{G}}$, monotone, upper-bounding matrices in the sense of the $\leq_{\text{st}}$ and the $\leq_{\text{icx}}$ orders. Class $\mathcal{C}^{\mathcal{G}}$ matrices are completely defined by means of three vectors: $\boldsymbol{v}$, $\boldsymbol{c}$, and $\boldsymbol{r}$. Our method consists of two steps. First, we use the information contained in the last column of the original transition matrix to

TABLE 1. Best $\leq_{\text{st}}$ and $\leq_{\text{icx}}$ Bounds Derived from Bounds of Figures 6 and 7

	$\text{Prob}(N \geq x)$		$E[(N-x+1)^+]$	
x	Exact	$\leq_{\text{st}}$ Upper bound (vector $\boldsymbol{r}$)	Exact	$\leq_{\text{icx}}$ Upper bound (vector $\boldsymbol{r}$)
5	6.6291146e−01	9.9995271e−01 (Class C)	3.4477356e + 00	5.6191482e + 00 (Class C)
10	2.5497376e−01	6.8694542e−01 (Class C)	1.0437843e + 00	2.5040774e + 00 (Icx-Simplex)
15	7.0334260e−02	3.2739325e−01 (St-LC)	2.0717100e−01	9.2096037e−01 (Class C)
20	1.0418200e−02	5.3203987e−02 (St-LC)	1.0418200e−02	3.3491086e−02 (Icx-Simplex)

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

TABLE 2. Best $\leq_{st}$ and $\leq_{icx}$ Transient Bounds for the Same Set of Parameters as in Figures 6 and 7

Time (slot)	Prob$(N \geq 15)$		$E[(N-14)^+]$	
	Exact	$\leq_{st}$ Upper bound (vector $\boldsymbol{r}$)	Exact	$\leq_{icx}$ Upper bound (vector $\boldsymbol{r}$)
1	7.0368744e−03	7.0368744e−03 (Class C)	7.0368744e−03	7.0368744e−03 (Class C)
2	3.0172134e−02	1.2322191e−01 (St-LC)	7.2388355e−02	3.1458996e−01 (Icx-Simplex)
5	6.3273645e−02	2.5414542e−01 (St-LC)	1.8249108e−01	7.2576011e−01 (Icx-Simplex)
10	7.0035603e−02	3.1412557e−01 (St-LC)	2.0612093e−01	9.1155294e−01 (Class C)
20	7.0333697e−02	3.2695794e−01 (St-LC)	2.0716904e−01	9.2090214e−01 (Class C)
50	7.0334200e−02	3.2739323e−01 (St-LC)	2.0717080e−01	9.2096037e−01 (Class C)

compute vector $\boldsymbol{r}$ of row increments. Once vector $\boldsymbol{r}$ is fixed, we apply Algorithm 1 ($\leq_{st}$ bounds) or Algorithm 2 ($\leq_{icx}$ bounds) to determine the remaining $\boldsymbol{v}$ and $\boldsymbol{c}$ vectors.

We present different choices of vector $\boldsymbol{r}$ to construct class $\mathcal{C}^{\mathcal{G}}$ bounding matrices in the sense of both $\leq_{st}$ and $\leq_{icx}$ orders. By changing input vector $\boldsymbol{r}$, it is possible to easily construct different bounding matrices in class $\mathcal{C}^{\mathcal{G}}$. These matrices can be analyzed through the closed-form solutions and the better bound can be taken for a given performance measure of interest.

We have performed numerical experiments to study the accuracy of bounds as a function of matrix structures and different choices of vector r. We observe that class $\mathcal{C}^{\mathcal{G}}$ bounds become tighter when the sparseness of the underlying matrix decreases and that $\leq_{icx}$ bounds are tighter in general than $\leq_{st}$ bounds. We show that $\leq_{st}$ bounds can be improved by class $\mathcal{C}^{\mathcal{G}}$ matrices compared to class $\mathcal{C}$ matrices. This is especially useful when we are interested in functionals of distributions that are increasing but not convex. For $\leq_{icx}$ bounds, we conclude that class $\mathcal{C}$ bounds are generally accurate. Other class $\mathcal{C}^{\mathcal{G}}$ bounds could be interesting when computing bounds for rewards defined on last states.

Computing closed-form stochastic bounds seems to be a promising approach to overcome the state space explosion problem of Markov chains. We are considering extending our work to look for other matrix structures having closed-form solutions and to other stochastic orders.

References

1. Abu-amsha, O. & Vincent, J.M. (1998). An algorithm to bound functionals of Markov chains with large state space. In *4th INFORMS Conference on Telecommunications*, Florida.
2. Ben Mamoun, M. (2002). Encadrements stochastiques et évaluation de performances des réseaux. PhD thesis, University of Versailles.
3. Ben Mamoun, M. & Pekergin, N. (2000). Computing closed-form stochastic bounds on the stationary distribution of Markov chains. In D. Gardy & A. Mokkadem (eds.), *Mathematics and computer science: Algorithms, trees, combinatorics and probabilities*. Basel: Birkhauser-Verlag, pp. 197–209.
4. Ben Mamoun, M. & Pekergin, N. (2002). Closed-form stochastic bounds on the stationary distribution of Markov chains. *Probability in the Engineering and Informational Sciences* 16: 403–426.

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

5. Ben Mamoun, M. & Pekergin, N. (2005). Computing closed-form stochastic bounds on transient distributions of Markov chains. In M. Papazoglu & K. Yamazaki (eds.), *Applications and the Internet SAINT2005 workshops*. New York: IEEE Computer Society, pp. 260–263.
6. Courtois, P.J. & Semal, P. (1984). Bounds for the positive eigenvectors of nonnegative matrices and their approximation by decomposition. *Journal of the ACM* 31(4): 804–825.
7. Fourneau, J.M. & Pekergin, N. (2002). An algorithmic approach to stochastic bounds. In M. Calzrossa & S. Tucci (eds.), *Performance evaluation of complex systems: Techniques and tools*. Berlin: Springer-Verlag, pp. 64–89.
8. Golubchik, L. & Lui, J. (1997). Bounding of performance measures for a threshold-based queuing systems with hysteresis. In *Proceeding of ACM SIGMETRICS'97*, pp. 147–157.
9. Keilson, J. & Kester, A. (1977). Monotone matrices and monotone Markov processes. *Stochastic Processes and their Applications* 5: 231–241.
10. Li, H. & Shaked, M. (1994). Stochastic convexity and concavity of Markov processes. *Mathematics in Operations Research*, 19(2): 477–493.
11. Lui, J., Muntz, R., & Towsley, D. (1998). Computing performance bounds of Fork–Join parallel programs under a multiprocessing environment. *IEEE Transactions on Parallel and Distributed Systems* 9(3): 295–311.
12. Mahevas, S. & Rubino, G. (2001). Bound computation of dependability and performance measures. *IEEE Transactions on Computers* 50(5): 399–413.
13. Muller, A. & Stoyan, D. (2002). *Comparison methods for stochastic models and risks*. New York: Wiley.
14. Muntz, R.R. & Lui, J. (1994). Computing bounds on steady-state availability of repairable computer systems. *Journal of the ACM* 41(4): 676–707.
15. Pekergin, N. (1999). Stochastic performance bounds by state space reduction. *Performance Evaluation* 36–37, 1–17.
16. Shaked, M. & Shantikumar, J.G. (1994). *Stochastic orders and their applications*. San Diego: Academic Press.
17. Stewart, W.J. (1994). *Introduction to the numerical solution of Markov chains*. Princeton, NJ: Princeton University Press.
18. van Dijk, N.M. (1998). Bounds and error bounds for queueuing networks. *Annals of Operations Research* 79: 295–319.
19. van Dijk, N.M. & van der Wal, J. (1989). Simple bounds and monotonicity results for finite multiserver exponential tandem queues. *Queueing Systems* 4: 1–16.

APPENDIX

In this appendix we give a common proof for Theorems 3 and 4. For that, we will first give some definitions to have a common notation for Algorithms 1 and 2 and related properties. In the sequel, $\leq_{\mathcal{F}}$ will denote either the $\leq_{\text{st}}$ order or the $\leq_{\text{icx}}$ order. It corresponds to the $\leq_{\text{st}}$ order in the context of Algorithm 1 and to the the $\leq_{\text{icx}}$ order in the context of Algorithm 2.

Let ϕ_{st} and $\tilde{\phi}_{\text{st}}$ be the operators $\phi_{\text{st}}: \mathbb{R}^n \times \mathbb{N} \to \mathbb{R}$ and $\tilde{\phi}_{\text{st}}: \mathbb{R}^n \times \mathbb{N} \to \mathbb{R}$, respectively, defined by

$$\phi_{\text{st}}(\boldsymbol{x}, j) = \sum_{k=j}^{n} x_k \quad \text{and} \quad \tilde{\phi}_{\text{st}}(\boldsymbol{x}, j) = \sum_{k=j+1}^{n} x_k. \tag{A1}$$

Note that lines 1, 2, 5, and 6 are the same in Algorithms 1 and 2. In addition, as $Q_{i,*} = \boldsymbol{v} + r_i \boldsymbol{c}$, lines 3 and 4 can be written in a common way as follows:

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

3. $v_j = \left[\max_{1 \le i < M} \dfrac{r_n \left(\phi_{\mathcal{F}}(P_{i,*}, j) - \tilde{\phi}_{\mathcal{F}}(Q_{i,*}, j) + \sum_{k=j+1}^{n} q_{i,k} \right) - r_i}{r_n - r_i} - \sum_{k=j+1}^{n} v_k \right]^+$;

4. $c_j = \max\left(-v_j / r_n, \left[\max_{K \le i \le n} \dfrac{\phi_{\mathcal{F}}(P_{i,*}, j) - \phi_{\mathcal{F}}(\boldsymbol{v}, j)}{r_i} \right]^+ - \tilde{\phi}_{\mathcal{F}}(\boldsymbol{c}, j) \right)$.

On the other hand, both conditions $\boldsymbol{r} \in \mathcal{F}_{\mathrm{st}}$ and $\boldsymbol{r} \in \mathcal{F}_{\mathrm{icx}}$, corresponding respectively to $\le_{\mathrm{st}}$ and $\le_{\mathrm{icx}}$ orders, imply that $\boldsymbol{r}$ is increasing. Moreover, the input vector $\boldsymbol{r}$ for both algorithms is nonnegative. Note also that the operators $\phi_{\mathcal{F}}$ and $\tilde{\phi}_{\mathcal{F}}$ are linear in the first component and that $\phi_{\mathcal{F}}(\boldsymbol{x}, j) = \tilde{\phi}_{\mathcal{F}}(\boldsymbol{x}, j) + x_j$ for both orders $\le_{\mathrm{st}}$ and $\le_{\mathrm{icx}}$. However, we do not have the same relation between $\phi_{\mathcal{F}}(\boldsymbol{x}, j)$ and $\phi_{\mathcal{F}}(\boldsymbol{x}, j+1)$, which is why we separate the cases $\le_{\mathrm{st}}$ and $\le_{\mathrm{icx}}$ for some points of the proof of Theorems 3 and 4. Let us remark that

$$\phi_{\mathrm{st}}(\boldsymbol{x}, j) = \phi_{\mathrm{st}}(\boldsymbol{x}, j+1) + x_j \quad \text{and} \quad \phi_{\mathrm{icx}}(\boldsymbol{x}, j) = \phi_{\mathrm{icx}}(\boldsymbol{x}, j+1) + \phi_{\mathrm{st}}(\boldsymbol{x}, j). \tag{A2}$$

For the sake of clarity of this proof, we give first the following two lemmas.

LEMMA A1: *Let $\boldsymbol{P}$ be a stochastic matrix. The matrix $\boldsymbol{Q} = \boldsymbol{e}^t \boldsymbol{v} + \boldsymbol{r}^t \boldsymbol{c}$ computed by Algorithm 1 or by Algorithm 2 satisfies*

$$\boldsymbol{P} \le_{\mathcal{F}} \boldsymbol{Q}; \quad \textit{i.e.,}\ \boldsymbol{P}\boldsymbol{K}_{\mathcal{F}} \le \boldsymbol{Q}\boldsymbol{K}_{\mathcal{F}} \Leftrightarrow \phi_{\mathcal{F}}(P_{i,*}, j) \le \phi_{\mathcal{F}}(Q_{i,*}, j), \quad \forall i, \forall j. \tag{A3}$$

PROOF:

Column $j = n$: We have $\phi_{\mathcal{F}}(P_{i,*}, n) = p_{i,n}$ and $\phi_{\mathcal{F}}(Q_{i,*}, n) = q_{i,n}$. For $1 \le i < K$, as $r_i = 0$, from line 1 it follows that $q_{i,n} = v_n + r_i c_n = v_n \ge p_{i,n}$. For $K \le i \le n$, line 2 and $\boldsymbol{r}$ nonnegative $\Rightarrow q_{i,n} = v_n + r_i c_n \ge p_{i,n}$.

Column j, $2 \le j \le n-1$: For $i < K$, $r_i = 0$ and line 3 of the algorithm $\Rightarrow q_{i,j} = v_j \ge \phi_{\mathcal{F}}(P_{i,*}, j) - \tilde{\phi}_{\mathcal{F}}(Q_{i,*}, j) \Rightarrow \phi_{\mathcal{F}}(Q_{i,*}, j) = q_{i,j} + \tilde{\phi}_{\mathcal{F}}(Q_{i,*}, j) \ge \phi_{\mathcal{F}}(P_{i,*}, j)$. For $K \le i \le n$, line 4 $\Rightarrow c_j \ge [(\phi_{\mathcal{F}}(P_{i,*}, j) - \phi_{\mathcal{F}}(\boldsymbol{v}, j))/r_i] - \tilde{\phi}_{\mathcal{F}}(\boldsymbol{c}, j) \Rightarrow \phi_{\mathcal{F}}(P_{i,*}, j) \le \phi_{\mathcal{F}}(Q_{i,*}, j) = \phi_{\mathcal{F}}(\boldsymbol{v}, j) + r_i \phi_{\mathcal{F}}(\boldsymbol{c}, j)$.

Column $j = 1$: Lines 5 and 6 imply that $\sum_{k=1}^{n} q_{i,k} = 1$. Thus, the case of the $\le_{\mathrm{st}}$ order is trivial ($\sum_{k=1}^{n} p_{i,k} = \sum_{k=1}^{n} q_{i,k} = 1$). For the $\le_{\mathrm{icx}}$ order, using (A2) we have $\phi_{\mathrm{icx}}(P_{i,*}, 1) = \phi_{\mathrm{icx}}(P_{i,*}, 2) + \sum_{k=1}^{n} p_{i,k} = \phi_{\mathrm{icx}}(P_{i,*}, 2) + 1 \le \phi_{\mathrm{icx}}(Q_{i,*}, 2) + 1 = \phi_{\mathrm{icx}}(Q_{i,*}, 2) + \sum_{k=1}^{n} q_{i,k} = \phi_{\mathrm{icx}}(Q_{i,*}, 1)$. ■

LEMMA A2: *Let $\boldsymbol{P}$ be a stochastic matrix and $\boldsymbol{Q} = \boldsymbol{e}^t \boldsymbol{v} + \boldsymbol{r}^t \boldsymbol{c}$ the matrix computed by Algorithm 1 or by Algorithm 2. Then we have the following:*

$$\boldsymbol{v} \textit{ is a stochastic vector,} \quad \textit{i.e.,}\ v_i \ge 0,\ \forall i, \boldsymbol{v}\boldsymbol{e}^t = 1, \tag{A4}$$

$$\boldsymbol{c}\boldsymbol{K}_{\mathcal{F}} \ge 0, \qquad \boldsymbol{c}\boldsymbol{e}^t = 0. \tag{A5}$$

PROOF: $\boldsymbol{c}\boldsymbol{e}^t = 0$ follows directly from line 6 of the algorithm. Line 2 of the algorithm assures $c_n \ge 0$ and line 4 gives $c_j \ge -\tilde{\phi}_{\mathcal{F}}(\boldsymbol{c}, j)$; hence, $\phi_{\mathcal{F}}(\boldsymbol{c}, j) \ge 0$, $\forall j > 1$. For $j = 1$, as $\sum_{k=1}^{n} c_k = \boldsymbol{c}\boldsymbol{e}^t = 0$, $\phi_{\mathrm{st}}(\boldsymbol{c}, 1) = 0$ and $\phi_{\mathrm{icx}}(\boldsymbol{c}, 1) = \phi_{\mathrm{icx}}(\boldsymbol{c}, 2) \ge 0$. We conclude that $\phi_{\mathcal{F}}(\boldsymbol{c}, j) \ge 0$, $\forall j$ (i.e., $\boldsymbol{c}\boldsymbol{K}_{\mathcal{F}} \ge 0$).

$\boldsymbol{v}\boldsymbol{e}^t = 1$ follows directly from line 5. By the construction of $\boldsymbol{v}$ (lines 1 and 3) we have $v_j \ge 0$, $\forall j > 1$. It remains us to show that $v_1 = 1 - \sum_{k=2}^{n} v_k \ge 0$ (i.e., $\sum_{k=2}^{n} v_k \le 1$). In order

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

to show this inequality, we will show that $\sum_{k=j}^{n} v_k \leq 1$ by induction on j, $2 \leq j \leq n$. Since $p_{i,n} \leq 1$, $1 \leq i < M$, from line 1 of the algorithm it follows that $v_n \leq 1$.

Let us suppose that $\sum_{k=j+1}^{n} v_k \leq 1$. From line 3, if $v_j = 0$, then $\sum_{k=j}^{n} v_k = \sum_{k=j+1}^{n} v_k \leq 1$. Otherwise,

$$v_j = \max_{1 \leq i < M} \frac{r_n \left(\phi_{\mathcal{F}}(P_{i,*}, j) - \tilde{\phi}_{\mathcal{F}}(Q_{i,*}, j) + \sum_{k=j+1}^{n} q_{i,k} \right) - r_i}{r_n - r_i} - \sum_{k=j+1}^{n} v_k.$$

In the case of the $\leq_{\text{st}}$ order, $\sum_{k=j}^{n} v_k = \max_{1 \leq i < M} [(r_n \sum_{k=j}^{n} p_{i,k} - r_i)/(r_n - r_i)] \leq 1$, since $\sum_{k=j}^{n} p_{i,k} \leq 1$ and $r_n > 0$.

In the case of the $\leq_{\text{icx}}$ order, as $\phi_{\text{icx}}(P_{i,*}, j+1) - \phi_{\text{icx}}(Q_{i,*}, j+1) \leq 0$ (Lemma A1),

$$\begin{aligned} \sum_{k=j}^{n} v_k &= \max_{1 \leq i < M} \frac{r_n(\phi_{\text{icx}}(P_{i,*}, j) - \phi_{\text{icx}}(Q_{i,*}, j+1)) - r_i}{r_n - r_i} \\ &= \max_{1 \leq i < M} \frac{r_n(\phi_{\text{icx}}(P_{i,*}, j+1) - \phi_{\text{icx}}(Q_{i,*}, j+1) + \sum_{k=j}^{n} p_{i,k}) - r_i}{r_n - r_i} \\ &\leq \max_{1 \leq i < M} \frac{r_n \sum_{k=j}^{n} p_{i,k} - r_i}{r_n - r_i} \leq 1. \end{aligned}$$

■

Proof of Theorems 3 and 4: From (A4) and the definition of $\boldsymbol{Q}$ as $\boldsymbol{Q} = \boldsymbol{e}^t \boldsymbol{v} + \boldsymbol{r}^t \boldsymbol{c}$, we have $\boldsymbol{Q} \in \mathcal{C}^{\mathcal{G}}$. Lemma A2 (relation A5) and Lemma A1 give that $\boldsymbol{Q}$ is monotone and an upper bound for $\boldsymbol{P}$ with respect to the order $\leq_{\mathcal{F}}$. It remains just to show that $\boldsymbol{Q}$ *is a stochastic matrix*.

From Lemma A2 we have $\boldsymbol{Q}_{i,*} \boldsymbol{e}^t = \boldsymbol{v} \boldsymbol{e}^t + r_i \boldsymbol{c} \boldsymbol{e}^t = 1$, $\forall i$. In order to show that $\boldsymbol{Q}$ is stochastic, we must show that $q_{i,j} \geq 0$, $\forall i, j$.

For the last column we have $v_n \geq 0$ and $c_n \geq 0$ (lines 1 and 2 of the algorithm), so since $\boldsymbol{r}$ is nonnegative, $q_{i,n} = v_n + r_i c_n \geq v_n \geq 0$, $\forall i$.

From line 4 of the algorithm we have $c_j \geq -(v_j / r_n) \Rightarrow q_{n,j} = v_j + c_j r_n \geq 0$, $2 \leq j \leq n-1$. As the vector $\boldsymbol{r}$ is increasing, all of the columns of $\boldsymbol{Q}$ are monotone (increasing for $c_j \geq 0$ and decreasing for $c_j < 0$) and contained between two values that are both nonnegative (v_j and $q_{n,j}$); thus, $q_{i,j} \geq 0$, $\forall i$, $2 \leq j \leq n-1$.

For the first column we have $q_{i,1} = 1 - \sum_{k=2}^{n} q_{i,k}$, $\forall i$, so we must show that $\sum_{k=2}^{n} q_{i,k} = \sum_{k=2}^{n} v_k + r_i \sum_{k=2}^{n} c_k \leq 1$, $\forall i$. If $\sum_{k=2}^{n} c_k \leq 0$, then $\sum_{k=2}^{n} q_{i,k} \leq \sum_{k=2}^{n} v_k \leq 1$, $\forall i$. In the other case ($\sum_{k=2}^{n} c_k > 0$), as $\boldsymbol{r}$ is increasing, we have $\sum_{k=2}^{n} q_{i,k} \leq \sum_{k=2}^{n} q_{n,k}$, $\forall i$, so it is sufficient to show that $\sum_{k=2}^{n} q_{n,k} \leq 1$. We will show it by induction on j. Let us remark that $r_i = r_n$ for $i \geq M$. For $j = n$ from line 1 of the algorithm we have

$$\begin{aligned} v_n \geq \frac{r_n p_{i,n} - r_i}{r_n - r_i}, \quad 1 \leq i < M &\Rightarrow \frac{p_{i,n} - v_n}{r_i} \leq \frac{1 - v_n}{r_n}, \quad K \leq i < M \\ &\Rightarrow c_n = \left[\max_{K \leq i \leq n} \frac{p_{i,n} - v_n}{r_i} \right]^+ \leq \frac{1 - v_n}{r_n} \qquad (\text{as } v_n \leq 1 \ (A4)) \\ &\Rightarrow q_{n,n} = v_n + r_n c_n \leq 1. \end{aligned}$$

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

Let us suppose that $\sum_{k=j+1}^{n} q_{n,k} \leq 1$. We deduce that

$$\sum_{k=j}^{n} v_k + r_n \sum_{k=j+1}^{n} c_k \leq 1 + v_j \Rightarrow \frac{-v_j}{r_n} \leq \frac{1 - \sum_{k=j}^{n} v_k}{r_n} - \sum_{k=j+1}^{n} c_k. \tag{A6}$$

On the other hand, note that $\tilde{\phi}_{\mathrm{st}}(\boldsymbol{c}, j) - \sum_{k=j+1}^{n} c_k = 0$ and $\tilde{\phi}_{\mathrm{icx}}(\boldsymbol{c}, j) - \sum_{k=j+1}^{n} c_k = \phi_{\mathrm{icx}}(\boldsymbol{c}, j+1) \geq 0$ (Lemma A2). Hence, $\tilde{\phi}_{\mathcal{F}}(\boldsymbol{c}, j) - \sum_{k=j+1}^{n} c_k \geq 0$. As $r_n > 0$ and $1 - \sum_{k=j}^{n} v_k \geq 0$ (from (A4)), we have

$$-\tilde{\phi}_{\mathcal{F}}(\boldsymbol{c}, j) \leq \frac{1 - \sum_{k=j}^{n} v_k}{r_n} - \sum_{k=j+1}^{n} c_k. \tag{A7}$$

From line 3 of the algorithm it follows that

$$\begin{aligned}
v_j \geq{}& \frac{r_n \left(\phi_{\mathcal{F}}(P_{i,*}, j) - \tilde{\phi}_{\mathcal{F}}(Q_{i,*}, j) + \sum_{k=j+1}^{n} q_{i,k} \right) - r_i}{r_n - r_i} - \sum_{k=j+1}^{n} v_k, \quad 1 \leq i < M \\
\Rightarrow{}& r_n \left[\phi_{\mathcal{F}}(P_{i,*}, j) - \tilde{\phi}_{\mathcal{F}}(Q_{i,*}, j) + \sum_{k=j+1}^{n} q_{i,k} - \sum_{k=j}^{n} v_k \right] \leq r_i \left(1 - \sum_{k=j}^{n} v_k \right), \quad 1 \leq i < M \\
\Rightarrow{}& r_n \left[\phi_{\mathcal{F}}(P_{i,*}, j) - \tilde{\phi}_{\mathcal{F}}(\boldsymbol{v}, j) - r_i \tilde{\phi}_{\mathcal{F}}(\boldsymbol{c}, j) + r_i \sum_{k=j+1}^{n} c_k - v_j \right] \leq r_i \left(1 - \sum_{k=j}^{n} v_k \right), \\
& \qquad 1 \leq i < M.
\end{aligned}$$

For $K \leq i < M$, $r_i > 0$, so we have

$$\frac{\phi_{\mathcal{F}}(P_{i,*}, j) - \phi_{\mathcal{F}}(\boldsymbol{v}, j)}{r_i} - \tilde{\phi}_{\mathcal{F}}(\boldsymbol{c}, j) \leq \frac{1 - \sum_{k=j}^{n} v_k}{r_n} - \sum_{k=j+1}^{n} c_k, \quad K \leq i < M. \tag{A8}$$

For $M \leq i \leq n$, $r_i = r_n$. In the case of the $\leq_{\mathrm{icx}}$ order, using the fact that $\phi_{\mathrm{icx}}(P_{i,*}, j+1) \leq \phi_{\mathrm{icx}}(Q_{i,*}, j+1) = \phi_{\mathrm{icx}}(\boldsymbol{v} + r_i \boldsymbol{c}, j+1)$ (Lemma A1) and relation (A2), we have

$$\begin{aligned}
& \frac{\phi_{\mathrm{icx}}(P_{i,*}, j+1) - \phi_{\mathrm{icx}}(\boldsymbol{v} + r_i \boldsymbol{c}, j+1) + \sum_{k=j}^{n} p_{i,k} - \sum_{k=j}^{n} v_k}{r_i} \leq \frac{1 - \sum_{k=j}^{n} v_k}{r_i} = \frac{1 - \sum_{k=j}^{n} v_k}{r_n} \\
& \Leftrightarrow \frac{\phi_{\mathrm{icx}}(P_{i,*}, j) - \phi_{\mathrm{icx}}(\boldsymbol{v}, j)}{r_i} - \phi_{\mathrm{icx}}(\boldsymbol{c}, j+1) - \sum_{k=j+1}^{n} c_k \leq \frac{1 - \sum_{k=j}^{n} v_k}{r_n} - \sum_{k=j+1}^{n} c_k \\
& \Leftrightarrow \frac{\phi_{\mathrm{icx}}(P_{i,*}, j) - \phi_{\mathrm{icx}}(\boldsymbol{v}, j)}{r_i} - \tilde{\phi}_{\mathrm{icx}}(\boldsymbol{c}, j) \leq \frac{1 - \sum_{k=j}^{n} v_k}{r_n} - \sum_{k=j+1}^{n} c_k.
\end{aligned}$$

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press

The last inequality is obviously satisfied in the case of the $\leq_{\mathrm{st}}$ order, since $\phi_{\mathrm{st}}(P_{i,*}, j) = \sum_{k=j}^{n} p_{i,k} \leq 1$ and $\tilde{\phi}_{\mathrm{st}}(\boldsymbol{c}, j) = \sum_{k=j+1}^{n} c_k$. We conclude that

$$\frac{\phi_{\mathcal{F}}(P_{i,*}, j) - \phi_{\mathcal{F}}(\boldsymbol{v}, j)}{r_i} - \tilde{\phi}_{\mathcal{F}}(\boldsymbol{c}, j) \leq \frac{1 - \sum_{k=j}^{n} v_k}{r_n} - \sum_{k=j+1}^{n} c_k, \quad M \leq i \leq n. \tag{A9}$$

From (A6)–(A9) it follows that $c_j \leq [(1 - \sum_{k=j}^{n} v_k)/r_n] - \sum_{k=j+1}^{n} c_k$, so $\sum_{k=j}^{n} q_{n,k} \leq 1$. By induction on j it follows that $\sum_{k=2}^{n} q_{n,k} \leq 1$, so we proved that $\boldsymbol{Q}$ is a stochastic matrix. ■

https://doi.org/10.1017/S0269964807070155 Published online by Cambridge University Press