Closed form absorption time bounds

Ana Bušić[(1)] and Nihal Pekergin[(1,2)]

[1] PRiSM, Université de Versailles-St-Quentin,
45, Av. des Etats-Unis, 78035 Versailles, France
[2] Laboratory Marin Mersenne, Université Paris 1, 75013 Paris, France

Abstract. We consider a class of Markov chains known for its closed form transient and steady-state distributions. We show that some absorbing chains can be also seen as members of this class and we provide the closed form solution for their absorption time distributions. By constructing upper and lower bounding chains that belong to this particular class one can easily compute both lower and upper bounds for absorption time distribution of an arbitrary absorbing Markov chain. We provide a new algorithm for the construction of bounding chains from this class and we show a possible application of these bounds.

1 Introduction

Markov chains are widely used to model complex systems due to their simplicity to represent in an intuitive manner the behavior of a studied system. Various high-level formalisms such as Stochastic Petri Nets, Stochastic Automata Networks or PEPA nets have been proposed making the modeling task more efficient since one needs only to specify different components of the system, their local behaviors and their interactions. The generation of the underlying Markov chain and the computation of different performance or reliability measures of interest can be then performed by one of many existing tools, depending on the formalism used. However, most of the resolution techniques for transient or steady-state distributions of Markov chains depend on the actual size of the entire state space, which is often near to the product of the number of states of each component of the system.

In order to overcome the state space explosion problem various approximation methods have been proposed which ignore or simplify the complicating aspects of the underlying models. However, only some of those methods estimate the error committed by the approximation or guarantee that the approximate value of measure of interest is smaller (resp. greater) than the exact one. Different bounding methods have been proposed for steady-state analysis. Most of them use linear algebra arguments and resume the steady-state analysis to the solution of a linear system. The advantage of stochastic comparison techniques is that they use probabilistic arguments, allowing thus both steady-state and transient analysis of the system.

Stochastic orderings have been largely studied in different areas of applied probability [13, 16]. Various stochastic orders have been proposed and the most

well known is certainly strong stochastic order defined through the comparison of the expectation of all increasing rewards. Different bounding methods have been proposed by using this stochastic order. The main idea behind all these methods is to modify the transitions of the original chain in order to simplify its analysis, yet preserving the comparison of both steady-state and transient distributions. The proof of this comparison can be established by different techniques such as sample-path arguments or stochastic monotonicity. While the first one is specific to the strong stochastic order, construction of monotone bounding chains can be used with different stochastic orders such as for instance increasing convex order allowing to compare the variability of two Markov chains. Moreover, it is possible to construct algorithmically a bounding monotone chain using both strong stochastic ordering [1] or increasing convex stochastic ordering [3]. Finally, the simplification of the model can consist in a special structure adapted to some special numerical resolution method [11, 9], in construction of lumpable bounding models reducing thus the size of the model [17] or in construction of the bounding models having known closed-form solutions that will be considered in this paper.

In [5, 6] a class of Markov chains having closed-form solution for the steady-state distribution has been introduced by Ben Mamoun and Pekergin: class $\mathcal{C}$ chains. The same authors showed that the chains from this class have also closed-form solutions for their transient distributions [7]. A generalization of this class to a larger $\mathcal{C}^{\mathcal{G}}$ class of Markov chains has been recently proposed in [4]. This larger class allows one more degree of freedom and permits thus to obtain more precise bounds. We show in this paper that this additional degree of freedom allows also simple construction of bounding matrices that belong to this new class. Class $\mathcal{C}^{\mathcal{G}}$ structures can be imposed as the bounding matrices, in the algorithmic stochastic comparison approach [6, 4]. Thus, bounds on distributions can be computed by means of closed-form solutions of class $\mathcal{C}^{\mathcal{G}}$ matrices leading to important numerical complexity reductions. For instance, in [8], class $\mathcal{C}$ bounds have been applied to perform a first step rapid model checking.

Several two-level resolution methods which consist in analyzing several smaller sub-models separately and then finding the global solution by combining sub-solutions have been proposed. In [10], the cycle time of PEPA model where each component is a PEPA sub-model is considered. Holding times of sub-models which are continuous Phase distributions are bounded by exponential random variables, providing thus a far simpler precedence PEPA model that can be then analyzed by classical numerical techniques. In reliability studies, bounding schemes have been proposed by dividing large state spaces in several macro-states. One macro-state is taken into account explicitly while the others are reduced to single states. The transition rates for these states are obtained by computing bounds on the underlying sub-models [14, 12]. Class $\mathcal{C}^{\mathcal{G}}$ bounds may be useful to compute bounds of sub-models. To illustrate the applicability of this approach, we consider in this work a task graph where task execution times are given by discrete time Phase distributions. We first compute $\mathcal{C}^{\mathcal{G}}$ class bounds for

task execution times and then incorporate them in the considered task graph which is the high-level model.

We present briefly in Section 2 the generalized class $\mathcal{C}$ matrices and their basic properties: closed-form solutions for transient and steady-state distributions. We show that some absorbing chains also belong to this class and we give the closed form solution for their absorption times. This class of matrices can therefore be used to compute simple lower or upper absorption time bounds. In Section 3 we provide a new algorithm for construction of bounds that belong to this class. This new algorithm presents different advantages compared to the algorithm presented in [4]. The most important is certainly its simplicity and the possibility of computing both upper and lower class $\mathcal{C}^{\mathcal{G}}$ bounds. Finally, we show in Section 4 how this new algorithm can be used to obtain fast closed form bounds for PH distributions modeling service times in a multi-level model.

2 Class $\mathcal{C}^{\mathcal{G}}$ matrices

The $\mathcal{C}^{\mathcal{G}}$ class of stochastic matrices has been introduced recently in [4] as the generalization of the $\mathcal{C}$ class introduced in [5, 6]. We give first the definition of the $\mathcal{C}^{\mathcal{G}}$ class and the closed form solution for transient and steady-state distributions for the transition matrices that belong to this class. Further details and proofs can be found in [4].

2.1 Definition and basic properties

Throughout this paper we will denote the vectors and the matrices with boldface letters. All the vectors are row vectors and $\boldsymbol{x}^t$ will denote column vector obtained by transposing the vector $\boldsymbol{x}$. When comparing vectors or matrices $\boldsymbol{x} \leq \boldsymbol{y}$ stands for the usual component-wise comparison: $\boldsymbol{x} \leq \boldsymbol{y} \Leftrightarrow x_i \leq y_i, \forall i$.

Definition 1 (Class $\mathcal{C}^{\mathcal{G}}$). *A stochastic matrix $\boldsymbol{P}$ of size n belongs to $\mathcal{C}^{\mathcal{G}}$ if there are three vectors $\boldsymbol{v}$, $\boldsymbol{c}$ and $\boldsymbol{r}$ in $\mathbb{R}^n$ such that:*

$$\boldsymbol{P} = \mathbf{1}^t \boldsymbol{v} + \boldsymbol{r}^t \boldsymbol{c},$$

where $\mathbf{1}$ denotes the row-vector having all the components equal to 1 and the vectors $\boldsymbol{v}$ and $\boldsymbol{c}$ satisfy:

- *$\boldsymbol{v}$ is a probability vector ($v_i \geq 0, \forall i$, $\boldsymbol{v}\mathbf{1}^t = \sum_{i=1}^n v_i = 1$),*
- *$\boldsymbol{c}\mathbf{1}^t = \sum_{i=1}^n c_i = 0$.*

For vector $\boldsymbol{r}$ such that $r_i = i - 1, \forall i$ we obtain the class $\mathcal{C}$ Markov chains defined in [5]. Thus the class $\mathcal{C}^{\mathcal{G}}$ is larger than the class $\mathcal{C}$.

Certainly one of most important properties of class $\mathcal{C}^{\mathcal{G}}$ matrices is that they have closed form solution for the transient distributions:

Theorem 1 (Transient distributions). *[4, Theorem 2] Let* $\{X_k,\ k \geq 0\}$ *be a discrete time Markov chain with probability transition matrix* $\boldsymbol{P}$ *that belongs to the class* $\mathcal{C}^{\mathcal{G}}$*, such that* $\boldsymbol{P} = \mathbf{1}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c}$*, and let* $\boldsymbol{\nu}^k$ *be the distribution vector of* X_k*. Let* α*,* β *and* γ *be the following constants:*

$$\alpha = \boldsymbol{c}\boldsymbol{r}^t, \quad \beta = \boldsymbol{v}\boldsymbol{r}^t \text{ and } \gamma = \boldsymbol{\nu}^0\boldsymbol{r}^t.$$

Then for all $k \geq 0$*,*

$$\boldsymbol{\nu}^k = \boldsymbol{v} + a_k\boldsymbol{c}, \tag{1}$$

where a_k *is the constant defined as:*

$$a_k = \beta\sum_{i=0}^{k-2}\alpha^i + \gamma\alpha^{k-1} = \begin{cases} \beta\ \frac{(1-\alpha^{k-1})}{1-\alpha} + \gamma\ \alpha^{k-1}, & \alpha \neq 1 \\ \beta\ (k-1) + \gamma, & \alpha = 1. \end{cases}$$

The following corollary gives now the closed-form form for the steady-state distribution:

Corollary 1 (Steady-state distribution). *[4, Corollary 1] Let* $\{X_k,\ k \geq 0\}$ *be a discrete time Markov chain with probability transition matrix* $\boldsymbol{P} \in \mathcal{C}^{\mathcal{G}}$ *such that* $\boldsymbol{P} = \mathbf{1}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c}$ *and let* $\alpha = \boldsymbol{c}\boldsymbol{r}^t$*. If* $|\alpha| < 1$*, then* $\{X_k,\ k \geq 0\}$ *has a steady-state distribution given by:*

$$\boldsymbol{\pi} = \boldsymbol{v} + R\boldsymbol{c}, \tag{2}$$

where R *is a constant defined as:* $R = \frac{\boldsymbol{v}\boldsymbol{r}^t}{1-\boldsymbol{c}\boldsymbol{r}^t}$*.*

2.2 Closed form solution for absorption time distribution

Note that one can easily construct absorbing transition matrices that belong to the class $\mathcal{C}^{\mathcal{G}}$. Indeed, take an arbitrary probability vector $\boldsymbol{v}$ and define the vector $\boldsymbol{c}$ as follows:

$$c_i = -v_i,\ i < n,\ c_n = 1 - v_n.$$

Then obviously $\boldsymbol{c}\mathbf{1}^t = 0$. Then for an arbitrary vector $\boldsymbol{r}$ satisfying:

$$0 \leq r_i \leq 1,\ \forall i,$$
$$r_n = 1,$$

the matrix is clearly stochastic, thus $\boldsymbol{P} \in \mathcal{C}^{\mathcal{G}}$. Furthermore, we have $\boldsymbol{P}_{n,*} = (0, 0, \ldots, 0, 1)$, so state n is absorbing.

Example 1. Let $\boldsymbol{v} = (0.1\ \ 0.3\ \ 0.6)$. Thus $\boldsymbol{c} = (-0.1\ \ -0.3\ \ 0.4)$. By taking $\boldsymbol{r} = (0.1\ \ 0.2\ \ 1)$, matrix $\boldsymbol{P} = \mathbf{1}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c}$ is defined as follows:

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0.1 & 0.3 & 0.6 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.2 \\ 1 \end{pmatrix} \begin{pmatrix} -0.1 & -0.3 & 0.4 \end{pmatrix} = \begin{pmatrix} 0.09 & 0.27 & 0.64 \\ 0.08 & 0.24 & 0.68 \\ 0 & 0 & 1 \end{pmatrix}$$

Suppose now that we have an absorbing discrete time Markov chain $\{X_k,\ k \geq 0\}$ such that the unique absorbing state is the last state (state n). Denote by $\boldsymbol{P}$ the transition matrix of this chain. We show now that class $\mathcal{C}^{\mathcal{G}}$ matrices have also the closed form solution for the absorption time.

Proposition 1. *Let T denote the absorption time of an absorbing discrete time Markov chain $\{X_k,\ k \geq 0\}$ such that the unique absorbing state is the last state (state n). Denote by $\boldsymbol{t} \in \mathbb{N}^{\infty}$ the probability distribution vector of T. If the transition matrix $\boldsymbol{P} \in \mathcal{C}^{\mathcal{G}}$ ($\boldsymbol{P} = \mathbf{1}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c}$), then the vector $\boldsymbol{t}$ satisfies:*

$$\begin{aligned} t_0 &= \nu_n^0, \\ t_1 &= v_n + \gamma c_n - \nu_n^0, \\ t_k &= \alpha^{k-2}(\beta + \gamma(\alpha - 1))c_n,\ k \geq 2, \end{aligned}$$

Proof. Note first that for each k, $P(T \leq k) = P(X_k = n)$. Thus $t_0 = P(T = 0) = \nu_n^0$ and

$$t_k = P(T = k) = P(T \leq k) - P(T \leq k-1) = \nu_n^k - \nu_n^{k-1},\ \forall k > 0,$$

where $\boldsymbol{\nu}^k$ denotes the distribution of X_k. From Theorem 1 we have

$$\begin{aligned} t_1 &= v_n + \gamma c_n - \nu_n^0, \\ t_k &= (\boldsymbol{v} + a_k\boldsymbol{c})_n - (\boldsymbol{v} + a_{k-1}\boldsymbol{c})_n = (a_k - a_{k-1})\ c_n \\ &= \beta\alpha^{k-2} + \gamma(\alpha^{k-1} - \alpha^{k-2})c_n = \alpha^{k-2}(\beta + \gamma(\alpha - 1))c_n,\ k \geq 2. \end{aligned}$$

□

Corollary 2. *If $|\alpha| < 1$ then the mean absorption time $E[T]$ is finite and equal to:*

$$E[T] = v_n + \left[\gamma + (\beta + \gamma(\alpha - 1))\frac{2 - \alpha}{(1 - \alpha)^2}\right] c_n.$$

Proof. From Proposition 1 we have :

$$\begin{aligned} E[T] &= v_n + \gamma c_n + \sum_{k=2}^{\infty} k t_k = v_n + \gamma c_n + (\beta + \gamma(\alpha - 1))c_n \sum_{k=2}^{\infty} k\alpha^{k-2} \\ &= v_n + \gamma c_n + (\beta + \gamma(\alpha - 1))c_n \left(2\sum_{i=0}^{\infty} \alpha^i + \sum_{i=0}^{\infty} i\alpha^i\right) \\ &= v_n + \gamma c_n + (\beta + \gamma(\alpha - 1))c_n \left(\frac{2}{1 - \alpha} + \frac{\alpha}{(1 - \alpha)^2}\right) = \\ &= v_n + \left[\gamma + (\beta + \gamma(\alpha - 1))\frac{2 - \alpha}{(1 - \alpha)^2}\right] c_n. \end{aligned}$$

□

3 Algorithmic construction of bounding matrices

The closed-form solutions to compute transient distributions and absorption time distribution make class $\mathcal{C}^{\mathcal{G}}$ matrices interesting to construct bounding chains. These bounds can be derived by means of stochastic comparison techniques. We first state the basic definitions and theorems of this approach and refer to [13, 16] for further details.

3.1 Stochastic Comparison

Let denote by $\mathcal{F}_{st}$ the class of all increasing real functions on $\mathcal{E}$ and by $\mathcal{F}_{icx}$ the class of all increasing and convex real functions on $\mathcal{E}$. We denote by $\preceq_{\mathcal{F}}$ the stochastic order relation, where $\mathcal{F}$ can be replaced by st or icx to be associated respectively to the class of functions $\mathcal{F}_{st}$ or $\mathcal{F}_{icx}$.

Definition 2. *Let X and Y be two random variables taking values on a totally ordered state space $\mathcal{E}$.*

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

whenever the expectations exist.

In the sequel, we consider $\mathcal{E} = \{1, \ldots, n\}$ with the usual total order $\leq$. Stochastic orders $\preceq_{st}$ and $\preceq_{icx}$ can be also defined through matrices. We give here $\boldsymbol{K_{st}}$ and $\boldsymbol{K_{icx}}$ matrices related respectively to the $\preceq_{st}$ and $\preceq_{icx}$ orders. In the sequel $\boldsymbol{K_{\mathcal{F}}}$ denotes the matrix related to the $\preceq_{\mathcal{F}}$ order, $\mathcal{F} \in \{st, icx\}$.

$$\boldsymbol{K_{st}} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 1 & 0 & \ldots & 0 \\ 1 & 1 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \ldots & 1 \end{pmatrix} \qquad \boldsymbol{K_{icx}} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 2 & 1 & 0 & \ldots & 0 \\ 3 & 2 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n & n-1 & n-2 & \ldots & 1 \end{pmatrix}$$

Notice that for discrete random variables X and Y with probability vectors $\boldsymbol{p}$ and $\boldsymbol{q}$, the notations $\boldsymbol{p} \preceq_{\mathcal{F}} \boldsymbol{q}$ and $X \preceq_{\mathcal{F}} Y$ are used interchangeably. Moreover, we have the following characterization [13]:

Property 1. $X \preceq_{\mathcal{F}} Y$ if and only if $\boldsymbol{pK_{\mathcal{F}}} \leq \boldsymbol{qK_{\mathcal{F}}}$.

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 stochastically the underlying chains. We first define the monotonicity and comparability of stochastic matrices and then state this theorem.

Definition 3. *Let $\boldsymbol{P}$ be a stochastic matrix. $\boldsymbol{P}$ is said to be stochastically $\preceq_{\mathcal{F}}$-monotone if for any probability vectors $\boldsymbol{p}$ and $\boldsymbol{q}$,*

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

Definition 4. *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 $\preceq_{\mathcal{F}}$ order ($\boldsymbol{P} \preceq_{\mathcal{F}} \boldsymbol{Q}$) if*

$$\boldsymbol{P}\ \boldsymbol{K_{\mathcal{F}}} \leq \boldsymbol{Q}\ \boldsymbol{K_{\mathcal{F}}}.$$

Note that this is equivalent to saying that $\boldsymbol{P} \preceq_{\mathcal{F}} \boldsymbol{Q}$, if

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

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

Theorem 2. *Let $\boldsymbol{P}$ (resp. $\boldsymbol{Q}$) be the probability transition matrix of the time-homogeneous Markov chain $\{X_k, k \geq 0\}$ (resp. $\{Y_k, k \geq 0\}$). If*

- *$X_0 \preceq_{\mathcal{F}} Y_0$,*
- *at least one of the probability transition matrices is monotone, that is, either $\boldsymbol{P}$ or $\boldsymbol{Q}$ is $\preceq_{\mathcal{F}}$-monotone,*
- *the transition matrices are comparable, that is, $\boldsymbol{P} \preceq_{\mathcal{F}} \boldsymbol{Q}$,*

then

$$X_k \preceq_{\mathcal{F}} Y_k \quad \forall k.$$

Let $X = \{X_k, k \geq 0\}$ and $Y = \{Y_k, k \geq 0\}$ be now two Markov chains with an absorbing state n. Then under the same conditions as in the above theorem we have also the $\preceq_{st}$-comparison of absorption times to n. We will denote by T^X (resp. by T^Y) the absorption time to n of chain X (resp. Y).

Corollary 3. *[3, Proposition 2.9] Let P and Q be the transition matrices of two Markov chains $X = \{X_k, k \geq 0\}$ and $Y = \{Y_k, k \geq 0\}$ with an absorbing state n. If $X_0 \preceq_{\mathcal{F}} Y_0$, $\boldsymbol{P}$ or $\boldsymbol{Q}$ is $\preceq_{\mathcal{F}}$-monotone, and $\boldsymbol{P} \preceq_{\mathcal{F}} \boldsymbol{Q}$, then*

$$T^Y \preceq_{st} T^X.$$

Note that we obtain the $\preceq_{st}$-comparison of absorption times even if the two chains are comparable only in the increasing convex ordering sense. Indeed, the above result is even more general and the ordering relation needs only to allow the comparison of the probabilities of being in state n (see [3] for further details). Note also that the absorption time is $\preceq_{st}$-larger for a smaller chain in the $\preceq_{\mathcal{F}}$-ordering sense. This might seem strange at a first sight. Intuitively, as we consider the absorption time to the last state, the larger chain goes faster to this state and its absorption time is thus smaller.

Monotonicity properties of $\mathcal{C}^{\mathcal{G}}$ matrices The sufficient conditions for the monotonicity of class $\mathcal{C}^{\mathcal{G}}$ matrices are given in terms of vectors $\boldsymbol{c}$ and $\boldsymbol{r}$.

Proposition 2. *[4, Proposition 2] A matrix $\boldsymbol{P} \in \mathcal{C}^{\mathcal{G}}$, $\boldsymbol{P} = \mathbf{1}^t \boldsymbol{v} + \boldsymbol{r}^t \boldsymbol{c}$ such that:*

$$\boldsymbol{c}\boldsymbol{K_{\mathcal{F}}} \geq 0 \quad \textit{and} \quad \boldsymbol{r} \in \mathcal{F}$$

is $\preceq_{\mathcal{F}}$-monotone.

3.2 Algorithms for upper and lower bounding class $\mathcal{C}^{\mathcal{G}}$ monotone matrices

In [4] an algorithm for construction of upper bounding monotone $\mathcal{C}^{\mathcal{G}}$-matrices has been proposed for $\preceq_{st}$ and $\preceq_{icx}$-orders. The proposed algorithm takes as input an arbitrary stochastic matrix $\boldsymbol{P}$ and a non-negative vector $\boldsymbol{r} \in \mathcal{F}$ ($\mathcal{F}$ denotes either st or icx), and it returns vectors $\boldsymbol{c}$ and $\boldsymbol{v}$ such that $\boldsymbol{Q} = \mathbf{1}^t\boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c} \in \mathcal{C}^{\mathcal{G}}$, $\boldsymbol{P} \preceq_{\mathcal{F}} \boldsymbol{Q}$ and $\boldsymbol{Q}$ is $\preceq_{\mathcal{F}}$-monotone. Some heuristics for choosing a vector $\boldsymbol{r}$ by using some information from the original matrix $\boldsymbol{P}$ have also been proposed. However, the algorithm proposed in [4] is far from being intuitive and it cannot be easily modified to compute lower bounds. Note that this is not a problem in the case of the $\preceq_{st}$ order since, due to the symmetry properties of this order, lower $\preceq_{st}$-bounds can be obtained by inversing the order on the states of the chain and then computing an upper bound. In the case of class $\mathcal{C}$ or $\mathcal{C}^{\mathcal{G}}$-bounds, $\preceq_{icx}$ order provides often considerably more precise results [6, 4]. Unfortunately, this order is not symmetric and the algorithms that compute upper $\preceq_{icx}$-bounds cannot be used. The algorithm proposed in [4] is a direct generalization of algorithms proposed in [5, 6] to $\mathcal{C}^{\mathcal{G}}$ class of matrices. We propose here a far more intuitive algorithm, more adapted to the $\mathcal{C}^{\mathcal{G}}$-structure. Furthermore, we propose the algorithms for both upper and lower bounds.

Let $\boldsymbol{P}$ be an arbitrary stochastic matrix. We take the first row of matrix $\boldsymbol{P}$ for the vector $\boldsymbol{v}$. In order to find a vector $\boldsymbol{c}$, we compute first a probability vector $\boldsymbol{x}$ that is greater in the $\preceq_{\mathcal{F}}$-ordering sense than all the rows of matrix $\boldsymbol{P}$. We will discuss this step of the algorithm more in details after presenting the general structure of the algorithm, since this step depends on the underlying stochastic order. We would like to obtain a bounding matrix $\boldsymbol{Q}$ such that $\boldsymbol{Q}_{1,*} = \boldsymbol{v}$ and $\boldsymbol{Q}_{n,*} = \boldsymbol{x}$. In order to assure this, we can take $\boldsymbol{Q} = \mathbf{1}^t\ \boldsymbol{v} + \boldsymbol{h}^t\boldsymbol{c}$, where $h_1 = 0$, $h_n = 1$. Then $\boldsymbol{Q}_{n,*} = \boldsymbol{v} + \boldsymbol{c} = \boldsymbol{x}$ defines completely the vector $\boldsymbol{c}$:

$$\boldsymbol{c} = \boldsymbol{x} - \boldsymbol{v}.$$

Notice that we have clearly $\boldsymbol{c}\boldsymbol{K}_{\mathcal{F}} \geq \mathbf{0}$, as $\boldsymbol{v} \preceq_{\mathcal{F}} \boldsymbol{x}$ by the construction of vectors $\boldsymbol{x}$ and $\boldsymbol{v}$. In order to satisfy $\boldsymbol{P} \preceq_{\mathcal{F}} \boldsymbol{Q}$ we need to compute a vector $\boldsymbol{h}$ such that:

$$\boldsymbol{P}_{i,*} \preceq_{\mathcal{F}} \boldsymbol{v} + h_i\boldsymbol{c} = \boldsymbol{Q}_{i,*},$$

i.e. $\boldsymbol{v}\boldsymbol{K}_{\mathcal{F}} + h_i\boldsymbol{c}\boldsymbol{K}_{\mathcal{F}} \geq \boldsymbol{P}_{i,*}\boldsymbol{K}_{\mathcal{F}}$. Note $\boldsymbol{w} = \boldsymbol{v}\boldsymbol{K}_{\mathcal{F}}$ and $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{K}_{\mathcal{F}}$. Since $\boldsymbol{z} = \boldsymbol{c}\boldsymbol{K}_{\mathcal{F}} \geq \mathbf{0}$, we can take:

$$h_i = \max_{j \mid z_j > 0} \frac{A_{i,j} - w_j}{z_j}.$$

We obtain a vector $\boldsymbol{h} \leq \mathbf{1}$ as $\boldsymbol{z} = \boldsymbol{w} + \boldsymbol{y}$, where $\boldsymbol{y} \geq \boldsymbol{A}_{i,*}$, $,\forall 1 \leq i \leq n$. It remains us to satisfy the monotonicity constraints for matrix $\boldsymbol{Q}$. If vector $\boldsymbol{h} \in \mathcal{F}$, then Proposition 2 and $\boldsymbol{c}\boldsymbol{K}_{\mathcal{F}} \geq \mathbf{0}$ implies that $\boldsymbol{Q}$ is $\preceq_{\mathcal{F}}$-monotone. Unfortunately, we do not always have $\boldsymbol{h} \in \mathcal{F}$. However, note that we can modify the entries of vector $\boldsymbol{h}$ as long as they stay smaller than 1. Indeed, for a vector $\boldsymbol{r}$ such that $\boldsymbol{h} \leq \boldsymbol{r} \leq \mathbf{1}$ the matrix $\tilde{\boldsymbol{Q}} = \mathbf{1}^t\ \boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c}$ is also a stochastic matrix such that $\boldsymbol{P} \preceq_{\mathcal{F}} \tilde{\boldsymbol{Q}}$. Thus we need to find a vector $\boldsymbol{r}$ satisfying:

– $\boldsymbol{r} \in \mathcal{F}$,
– $\boldsymbol{h} \leq \boldsymbol{r} \leq \mathbf{1}$.

The construction of such a vector for the case of strong stochastic order and increasing convex order will be discussed later. The construction of upper bounding $\preceq_{\mathcal{F}}$-monotone $\mathcal{C}^{\mathcal{G}}$-matrix is given in Algorithm 1. The proof of the following theorem follows directly from the above discussion.

Algorithm 1: Construction of an $\preceq_{\mathcal{F}}$-monotone class $\mathcal{C}^{\mathcal{G}}$ upper bound

1 Set $\boldsymbol{v} = \boldsymbol{P}_{1,*}$ and set $\boldsymbol{w} = \boldsymbol{v}\ \boldsymbol{K}_{\mathcal{F}}$.
2 Find a probability vector $\boldsymbol{x}$ such that $\boldsymbol{P}_{i,*} \preceq_{\mathcal{F}} \boldsymbol{x}$, $\forall 1 \leq i \leq n$. Set $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{K}_{\mathcal{F}}$.
3 Compute $\boldsymbol{z} = \boldsymbol{y} - \boldsymbol{w}$ and $\boldsymbol{c} = \boldsymbol{z}\ \boldsymbol{K}_{\mathcal{F}}^{-1}$.
4 Let $\boldsymbol{A} = \boldsymbol{P}\ \boldsymbol{K}_{\mathcal{F}}$. Compute the vector $\boldsymbol{h} = (h_1, \ldots, h_n)$:

$$h_i = \max_{j \mid z_j > 0} \frac{A_{i,j} - w_j}{z_j}.$$

Note that we have always $h_1 = 0$.
5 Find a vector $\boldsymbol{r}$ such that $\boldsymbol{h} \leq \boldsymbol{r} \leq \mathbf{1}$ (component-wise) and $\boldsymbol{r} \in \mathcal{F}$.
6 Set $\boldsymbol{Q} = \mathbf{1}^t\ \boldsymbol{v} + \boldsymbol{r}^t\boldsymbol{c}$.

Theorem 3. *The matrix $\boldsymbol{Q}$ obtained by Algorithm 1 is a stochastic matrix that belongs to the class $\mathcal{C}^{\mathcal{G}}$. Moreover, the matrix $\boldsymbol{Q}$ is $\preceq_{\mathcal{F}}$-monotone and $\boldsymbol{P} \preceq_{\mathcal{F}} \boldsymbol{Q}$.*

A lower-bounding $\preceq_{\mathcal{F}}$-monotone matrix $\boldsymbol{Q} \in \mathcal{C}^{\mathcal{G}}$ can be obtained by a similar algorithm. Here we preserve the last row of the original matrix and we compute a probability vector $\boldsymbol{v}$ that is smaller in the $\preceq_{\mathcal{F}}$-sense than all the rows in the original matrix $\boldsymbol{P}$:

$$\boldsymbol{v} \preceq_{\mathcal{F}} \boldsymbol{P}_{i,*}.$$

In order to obtain a bounding matrix $\boldsymbol{Q}$ such that $\boldsymbol{Q}_{1,*} = \boldsymbol{v}$ and $\boldsymbol{Q}_{n,*} = \boldsymbol{P}_{n,*}$ we can take $\boldsymbol{Q} = \mathbf{1}^t\ \boldsymbol{v} + \boldsymbol{h}^t\boldsymbol{c}$, where $h_1 = 0$, $h_n = 1$ and $\boldsymbol{c} = \boldsymbol{P}_{n,*} - \boldsymbol{v}$. As $\boldsymbol{v} \preceq_{\mathcal{F}} \boldsymbol{P}_{n,*}$, we have $\boldsymbol{c}\boldsymbol{K}_{\mathcal{F}} = \boldsymbol{P}_{n,*} - \boldsymbol{v}\boldsymbol{K}_{\mathcal{F}} \geq \mathbf{0}$. To guarantee the comparison of the matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ we compute a vector $\boldsymbol{h}$ such that:

$$\boldsymbol{Q}_{i,*} = \boldsymbol{v} + h_i\boldsymbol{c} \preceq_{\mathcal{F}} \boldsymbol{P}_{i,*},$$

i.e. $\boldsymbol{v}\boldsymbol{K}_{\mathcal{F}} + h_i\boldsymbol{c}\boldsymbol{K}_{\mathcal{F}} \leq \boldsymbol{P}_{i,*}\boldsymbol{K}_{\mathcal{F}}$. Let us denote by $\boldsymbol{w} = \boldsymbol{v}\boldsymbol{K}_{\mathcal{F}}$ and $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{K}_{\mathcal{F}}$. Since $\boldsymbol{z} = \boldsymbol{c}\boldsymbol{K}_{\mathcal{F}} \geq \mathbf{0}$, we can take:

$$h_i = \min_{j \mid z_j > 0} \frac{A_{i,j} - w_j}{z_j}.$$

Note that by the construction of vector $\boldsymbol{v}$ as a vector that is smaller than all the rows of matrix $\boldsymbol{P}$, we clearly have: $\boldsymbol{h} \geq \mathbf{0}$ and $h_n = 1$. If the vector $\boldsymbol{h} \in \mathcal{F}$,

then Proposition 2 and $\boldsymbol{cK}_{\mathcal{F}} \geq \mathbf{0}$ imply that the matrix $\boldsymbol{Q}$ is $\preceq_{\mathcal{F}}$-monotone. As unfortunately this is not always the case, we need to modify this vector and, in order to preserve the comparison of the matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ and the stochastic property of matrix $\boldsymbol{Q}$, we can only decrease the entries of the vector $\boldsymbol{h}$. Thus, in order to satisfy the monotonicity constraints for matrix $\boldsymbol{Q}$ we need to compute a vector $\boldsymbol{r}$ such that:

- $\boldsymbol{r} \in \mathcal{F}$,
- $\mathbf{0} \leq \boldsymbol{r} \leq \boldsymbol{h}$.

We resume the construction of a lower bounding monotone $\mathcal{C}^{\mathcal{G}}$ matrix in Algorithm 2. We have shown the following theorem:

Theorem 4. *The matrix $\boldsymbol{Q}$ obtained by Algorithm 2 is a stochastic matrix that belongs to the class $\mathcal{C}^{\mathcal{G}}$. Moreover, the matrix $\boldsymbol{Q}$ is $\preceq_{\mathcal{F}}$-monotone and $\boldsymbol{Q} \preceq_{\mathcal{F}} \boldsymbol{P}$.*

Algorithm 2: Construction of an $\preceq_{\mathcal{F}}$-monotone class $\mathcal{C}^{\mathcal{G}}$ lower bound

1 Find a probability vector $\boldsymbol{v}$ such that $\boldsymbol{v} \preceq_{\mathcal{F}} \boldsymbol{P}_{i,*}$, $\forall 1 \leq i \leq n.$, Set $\boldsymbol{w} = \boldsymbol{vK}_{\mathcal{F}}$.
2 Compute $\boldsymbol{c} = \boldsymbol{P}_{n,*} - \boldsymbol{v}$, $\boldsymbol{z} = \boldsymbol{cK}_{\mathcal{F}}$.
3 A Let $\boldsymbol{A} = \boldsymbol{P}\ \boldsymbol{K}_{\mathcal{F}}$. Compute the vector $\boldsymbol{h} = (h_1, \ldots, h_n)$:

$$h_i = \min_{j \mid z_j > 0} \frac{A_{i,j} - w_j}{z_j}.$$

Note that we have always $h_n = 1$.
4 Find a vector $\boldsymbol{r}$ such that $\mathbf{0} \leq \boldsymbol{r} \leq \boldsymbol{h}$ (component-wise) and $\boldsymbol{r} \in \mathcal{F}$.
5 Set $\boldsymbol{Q} = \mathbf{1}^t\ \boldsymbol{v} + \boldsymbol{r}^t \boldsymbol{c}$.

Computation of an upper or lower bounding vector for all the rows of the original matrix. We discuss now the construction of an upper bounding vector $\boldsymbol{x}$ (resp. a lower bounding vector $\boldsymbol{v}$) for all the rows of the original matrix $\boldsymbol{P}$ in line 2 of Algorithm 1 (resp. line 1 of Algorithm 2) for the strong stochastic order and increasing convex order. The construction is similar for both orders, thus we will denote by $\boldsymbol{K}_{\mathcal{F}}$ the corresponding matrix $\boldsymbol{K}_{\mathcal{F}} = \boldsymbol{K}_{\boldsymbol{st}}$ (resp. $\boldsymbol{K}_{\mathcal{F}} = \boldsymbol{K}_{\boldsymbol{icx}}$). Let $\boldsymbol{A} = \boldsymbol{PK}_{\mathcal{F}}$. Then the upper bounding vector $\boldsymbol{x}$:

$$\boldsymbol{P}i, * \preceq_{\mathcal{F}} \boldsymbol{x},$$

can be obtained as $\boldsymbol{x} = \boldsymbol{y}\boldsymbol{K}_{\mathcal{F}}^{-1}$, where:

$$y_j = \max_{1 \leq i \leq n} A_{i,j},\ \forall j.$$

We have clearly $\boldsymbol{P}i, * \preceq_{\mathcal{F}} \boldsymbol{x}$. It remains us to show that the vector $\boldsymbol{x}$ is stochastic. We will show first that $\sum_{i=1}^{n} x_i = 1$. For strong stochastic order this is trivial since $\sum_{i=1}^{n} x_i = y_1 = \max_{1 \leq i \leq n} A_{i,1} = 1$, as $A_{i,1} = 1$, $\forall i$. For increasing convex order notice that $A_{i,1} = A_{i,2} + 1$, $\forall i$, thus $\sum_{i=1}^{n} x_i = y_1 - y_2 = 1$. Finally,

we need to show that $x_i \geq 0,\ \forall i$. For strong stochastic order we have clearly $1 = y_1 \geq y_2 \geq \ldots \geq y_n \geq 0$, thus $x_n = y_n \geq 0$ and $x_j = y_j - y_{j+1} \geq 0,\ j < n$. For increasing convex order the proof is slightly more complex. Note that for an arbitrary vector $\boldsymbol{a}$ we have

$$(\boldsymbol{aK_{icx}})_j = (\boldsymbol{aK_{st}})_j + (\boldsymbol{aK_{icx}})_{j+1},\ j < n, \tag{3}$$

Let $\boldsymbol{z} = \boldsymbol{xK_{st}}$. Note that we have also:

$$\boldsymbol{z} = \boldsymbol{y}\boldsymbol{K_{st}}^{-1}.$$

By using (3) it is now easy to show that the vector $\boldsymbol{z}$ satisfies:

$$1 = z_1 \geq z_2 \geq \ldots \geq z_n \geq 0.$$

Therefore, $x_j \geq 0,\ \forall j$ and vector $\boldsymbol{x}$ is stochastic. Note that this vector $\boldsymbol{x}$ is the smallest upper bounding vector for all the rows of matrix $\boldsymbol{P}$.

Similarly, a lower bounding vector $\boldsymbol{v}$ can be obtained as $\boldsymbol{v} = \boldsymbol{wK_{\mathcal{F}}}$ where:

$$w_j = \min_{1 \leq i \leq n} A_{i,j},\ \forall j.$$

The vector $\boldsymbol{v}$ is the greatest lower bounding vector for rows of matrix $\boldsymbol{P}$. The proof that the vector $\boldsymbol{v}$ is a stochastic vector is similar to the proof for the upper bounding case.

Computation of monotone bounding vectors. It remains us to show how to compute the upper or lower bounding monotone vectors in line 5 of Algorithm 1 and line 4 of Algorithm 2. We consider again strong stochastic and increasing convex order.

Strong stochastic order. In the upper bound case we need to find an increasing vector $\boldsymbol{r}$ such that $\boldsymbol{h} \leq \boldsymbol{r} \leq \boldsymbol{1}$. Additionally we know that $h_1 = 0$ and $h_i \leq 1,\ \forall i$ by the construction of vector $\boldsymbol{h}$. Therefore the vector $\boldsymbol{r}$, defined as $r_i = max_{k \leq i} h_i,\ \forall i$ and computed by:

$$r_1 = h_1,\ r_i = \max\{h_i,\ r_{i-1}\},\ i > 0,$$

satisfies clearly $\boldsymbol{h} \leq \boldsymbol{r} \leq \boldsymbol{1}$.

Similarly, in the case of a lower bound we have $h_n = 1$ and $\boldsymbol{h} \geq \boldsymbol{0}$ and a vector $\boldsymbol{r}$ such that $\boldsymbol{0} \leq \boldsymbol{r} \leq \boldsymbol{h}$ can be obtained by taking $r_i = min_{k \leq i} h_i,\ \forall i$, i.e.

$$r_n = h_n,\ r_i = \min\{h_i,\ r_{i+1}\},\ i < n.$$

Increasing convex order. Consider first the upper bounding case. We will suppose that the vector $\boldsymbol{h}$ is increasing. Note that if this is not the case, then an increasing vector $\boldsymbol{h'}$ such that $\boldsymbol{h} \leq \boldsymbol{h'} \leq \boldsymbol{1}$ should be first computed as described in the strong stochastic upper bound case described above. Then we need to find a

vector $\boldsymbol{r}$ such that $\boldsymbol{h'} \leq \boldsymbol{r} \leq \mathbf{1}$. For an increasing vector $\boldsymbol{h}$, an increasing and convex vector $\boldsymbol{r}$ such that $\boldsymbol{h} \leq \boldsymbol{r} \leq \mathbf{1}$ can then be easily obtained as follows:

$$r_n = h_n,\ r_{n-1} = h_{n-1},\ r_i = \max\{h_i,\ 2r_{i+1} - r_{i+2}\},\ i \leq n-2.$$

Let us now consider the lower bound computation. We need to find a vector $\boldsymbol{r}$ that is increasing and convex and that $\mathbf{0} \leq \boldsymbol{r} \leq \boldsymbol{h}$. Similarly as in the upper bounding case, we can suppose that the vector $\boldsymbol{h}$ is increasing. If this is not the case, we can find an increasing vector $\boldsymbol{h'}$ such that $\mathbf{0} \leq \boldsymbol{h'} \leq \boldsymbol{h}$ as described in lower bound computation case for the strong stochastic order. For an increasing vector $\boldsymbol{h}$ an increasing and convex vector $\boldsymbol{r}$ such that $\mathbf{0} \leq \boldsymbol{r} \leq \boldsymbol{h}$ can be find by Algorithm 3. We illustrate this Algorithm on an example in Figure 1.

Algorithm 3: Computation of lower bounding increasing convex vector

Notation : We will denote by $1 < s_1, s_2, \ldots, s_m \leq n$ the indexes for which the vector $\boldsymbol{h}$ strictly increases. Then we define $z_0 = 1$, and for i such that $1 \leq i \leq m$ we define the last index just before the vector $\boldsymbol{h}$ strictly increases: $z_i = s_i - 1,\ 1 \leq i \leq m$. Finally, if $z_m < n$ then we define $z_{m+1} = n$. For example for $(0.1, 0.3, 0.3, 0.5, 0.6)$ we have: $s_1 = 2, s_2 = 4, s_3 = 5$ and $z_0 = 1, z_1 = 1, z_2 = 3, z_3 = 4, z_5 = 5$.

```
1 i = 0,   r_1 = h_1
2 while (z_i < n) do
3     a = min_{j>i} (h_{z_j} - h_{z_i}) / (z_j - z_i),   k = arg min_{j>i} (h_{z_j} - h_{z_i}) / (z_j - z_i)
4     for (u = z_i + 1 to z_k) do  r_u = a u + r_{z_i}
5     i = k
6 end
```

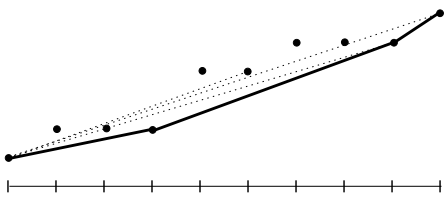

Fig. 1. Computation of lower bounding increasing convex vector by Algorithm 3. The bounding vector is constructed par intervals, by taking the smallest slope at each step.

Remark 1 (Complexity of Algorithms 1 and 2). The complexity of Algorithms 1 and 2 is quadratic with the size of the state space in the case of the full matrix implementation. Indeed, we used a matrix notation in both algorithms to simplify the presentation. Note that for instance $(\boldsymbol{PK_{st}})_{i,j}$ can be simply obtained as $(\boldsymbol{PK_{st}})_{i,j+1} + P_{i,j},\ j < n$ where $(\boldsymbol{PK_{st}})_{i,n} = P_{i,n}$. $\boldsymbol{PK_{icx}}$

can be easily computed using the fact that $\boldsymbol{K_{icx}} = \boldsymbol{K_{st}}^2$. In a similar way, $(\boldsymbol{AK}_{\mathcal{F}}^{-1})_{i,j}$ can be easily computed by applying the inverse transformation. For $\preceq_{st}$: $(\boldsymbol{AK_{st}}^{-1})_{i,j} = \boldsymbol{A}_{i,j} - \boldsymbol{A}_{i,j+1},\ j < n$, where $(\boldsymbol{AK_{st}}^{-1})_{i,j} = \boldsymbol{A}_{i,j}$.

Example 2. We illustrate here the $\mathcal{C}^{\mathcal{G}}$ absorption time bounds for mean absorption time (Corollary 2). Note however that Algorithms 1 and 2 can also be used to compute both transient and steady-state bounds. We consider a very simple example of an absorbing chain with n states, where state n is absorbing. We suppose that the initial state is 1 and for each state $i < n$ we have the following transitions:

- with probability a_i the system goes directly to the absorbing state n,
- with probability b_i the system goes to the next state $(i + 1)$,
- with probability $c_i = 1 - a_i - b_i$ the system returns to state 1.

Although the class $\mathcal{C}^{\mathcal{G}}$ bounds become interesting only for huge chains for which we cannot directly compute the absorption times using the classical numerical methods, we will consider here a small state space in order to easily compare the bounds with the exact values. In Table 1 we give the exact values and the bounds for the following parameters ($p = 0.6,\ n = 20$):

- Case A: $a_i = \frac{p}{2},\ b_i = 1 - p,\ c_i = \frac{p}{2},\ \forall i < n.$
- Case B: $a_i = \frac{pi}{n},\ b_i = 1 - p,\ c_i = \frac{p(n-i)}{n},\ \forall i < n.$

We can see from this example that it is not possible to compare the accuracy of the new upper bounds (providing lower bounds for absorption time) with those of [4]: they may be better or worse depending on the parameter values. We can see also that for this example $\preceq_{icx}$ bounds are more accurate than the $\preceq_{st}$ bounds as it is usually the case with class $\mathcal{C}$ or $\mathcal{C}^{\mathcal{G}}$ bounds.

Table 1. Class $\mathcal{C}^{\mathcal{G}}$ bounds for mean absorption time

	exact	st-inf (new)	st-inf [4]	st-sup	icx-inf (new)	icx-inf [4]	icx-sup
A	3.3333	2.3243	2.3243	3.3333	3.2386	2.5556	3.3333
B	1.8182	1.4704	1.7544	16.6667	1.7544	1.7544	3.4379

4 Bounding PH-distributions modeling service times

We consider a task graph with n nodes representing tasks where arcs represent synchronization constraints. The task execution (service) times are defined by discrete time PHase (PH) distributions. Let d_i (resp. t_i) be the execution time (resp. the completion) time of task (node) i and $Preced(i)$ be the set of immediate predecessors of i. Since task i can start its execution once all the predecessors have completed, task i terminates its execution at time t_i :

$$t_i = d_i + max_{j \in Preced(i)} t_j$$

Without loss of generality, we assume that task 1 has no predecessor and task n has no successor. Therefore $t_1 = d_1$ and t_n is the completion time of the task graph which is the measure of interest.

The absorbing chains representing discrete PH distributions constitute the low-level formalism while the task graph formalism is the high level formalism. Let us remark here that high level formalism can be extended to any (max,+) formalism [2].

The state space size of the Markov chain to compute the completion time grows exponentially with the number of tasks even for exponential (geometric) execution times. The stochastic bounds on the execution times of acyclic task graphs have been proposed in [15]. These bounds are based on the compatibility of the $\preceq_{icx}$ order with the *max* and the + operators. It has been proven that if $d_i^{inf} \preceq_{icx} d_i \preceq_{icx} d_i^{sup} \ \forall i$, then the completion time of the same task graph by considering bounding execution times, provides bounds on the completion time: $t_n^{inf} \preceq_{icx} t_n \preceq_{icx} t_n^{sup}$. Thus we propose to compute class $\mathcal{C}^{\mathcal{G}} \preceq_{icx}$ bounds on d_i which can be computed by the close-form solution of absorbing time given in section 2.2.

In the high-level model, bounds are provided by considering specific distributions with the same mean for task execution times. Lower bounds are computed by deterministic random variables while upper bounds are computed by geometric random variables. We do not give the proof here but refer to [15] for bounds on task graphs. The lower bound is well-known as *folk theorem*: deterministic minimizes the randomness [2]. The upper bound is established for a family of distributions used in reliability [13]. An integer valued X is called Discrete New Better than Used (DNBU), if $[X - t | X > t] \preceq_{st} X, \forall t$; X is called DNBUE if this is satisfied for the expectations: $E[X - t | X > t] \leq E[X], \forall t$. Geometric distributions are the maximal distributions for DNBUE distributions: If X is DNBUE of mean m, then X is smaller in the $\preceq_{icx}$ sense than geometric distributed random variable of mean m ($X \preceq_{icx} Geom(m)$). In [3], it has been shown that monotone PH distributions belong to DNBU distributions. Therefore d_i^{sup} can be remplaced by geometric distributions with the same means to provide upper bounds.

5 Conclusion

We have shown in this paper that the class $\mathcal{C}^{\mathcal{G}}$ matrices can be used to derive rapid bounds for absorption times. We proposed simple numerical algorithms to construct both lower and upper $\preceq_{st}$-(resp. $\preceq_{icx}$-)monotone bounds that belong to this class. To the best of our knowledge this is the first algorithm for lower $\preceq_{icx}$-monotone bound computation. For the simplicity of presentation, we consider here only discrete Markov chains. Similar results can be obtained for uniformizable continuous time Markov chains by applying the Algorithms 1 and 2 to the uniformized chain.

References

1. O. Abu-Amsha and J.-M. Vincent. An algorithm to bound functionals of markov chains with large state space. In *4th INFORMS Conference on Telecommunications, Boca Raton, FL*, 1998.
2. F. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems*. Willey, New York, NY, 1992.
3. M. Ben Mamoun, A. Busic, J.-M. Fourneau, and N. Pekergin. Increasing convex monotone markov chains: Theory, algorithm and applications. In *MAM 2006: Markov Anniversary Meeting*, pages 189–210, Raleigh, North Carolina, USA, 2006. Boson Books.
4. M. Ben Mamoun, A. Busic, and N. Pekergin. Generalized class C Markov chains and computation of closed-form bounding distributions. *Probability in the Engineering and Informational Sciences*, 21(2):235–260, 2007.
5. M. Ben Mamoun and N. Pekergin. Computing closed-form stochastic bounds on the stationary distribution of Markov chains. In *Mathematics and Computer Science: Algorithms, Trees, Combinatorics and Probabilities*, pages 197–209, Basel, 2000. Birkhauser Verlag.
6. M. Ben Mamoun and N. Pekergin. Closed-form stochastic bounds on the stationary distribution of Markov chains. *Probability in the Engineering and Informational Sciences*, 16(4):403–426, 2002.
7. M. Ben Mamoun and N. Pekergin. Computing closed-form stochastic bounds on transient distributions of Markov chains. In *SAINT-W '05: Proceedings of the 2005 Symposium on Applications and the Internet Workshops (SAINT 2005 Workshops)*, pages 260–263, Washington, DC, USA, 2005. IEEE Computer Society.
8. M. Ben Mamoun, N. Pekergin, and S. Younès. Model checking of continuous-time Markov chains by closed-form bounding distributions. In *QEST*, pages 189–198. IEEE Computer Society, 2006.
9. A. Busic and J.-M. Fourneau. A matrix pattern compliant strong stochastic bound. In *SAINT-W '05: Proceedings of the 2005 Symposium on Applications and the Internet Workshops (SAINT 2005 Workshops)*, pages 256–259, Washington, DC, USA, 2005. IEEE Computer Society.
10. J.-M. Fourneau and L. Kloul. A precedence pepa model for performance and reliability analysis. In *Formal Methods and Stochastic Models for Performance Evaluation, EPEW'06 Proceedings*, volume 4054 of *LNCS*, pages 1–15, Berlin Heidelberg, 2006. Springer-Verlag.
11. J.-M. Fourneau and N. Pekergin. An algorithmic approach to stochastic bounds. In *Performance Evaluation of Complex Systems: Techniques and Tools, Performance 2002, Tutorial Lectures*, pages 64–88, London, UK, 2002. Springer-Verlag.
12. S. Mahevas and G. Rubino. Bound computation of dependability and performance measures. *IEEE Trans. Comput.*, 50(5):399–413, 2001.
13. A. Muller and D. Stoyan. *Comparison Methods for Stochastic Models and Risks*. Wiley, New York, NY, 2002.
14. R. R. Muntz, E. de Souza e Silva, and A. Goyal. Bounding availability of repairable computer systems. *IEEE Trans. on Computers*, 38(12):1714–1723, 1989.
15. N. Pekergin and J.-M. Vincent. Stochastic bounds on execution times of parallel programs. *IEEE Trans. Softw. Eng.*, 17(10):1005–1012, 1991.
16. M. Shaked and J. G. Shantikumar. *Stochastic Orders*. Springer, 2007.
17. L. Truffet. Reduction technique for discrete time Markov chains on totally ordered state space using stochastic comparisons. *Journal of Applied Probability*, 37(3):795–806, 2000.