Theoretical Computer Science 452 (2012) 12–20

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

An algorithm approach to bounding aggregations of multidimensional Markov chains

Hind Castel-Taleb [a,*], Lynda Mokdad [b], Nihal Pekergin [b]

[a] Telecom Sudparis/SAMOVAR, 9, rue Charles Fourier, 91011 Evry Cedex, France

[b] LACL, Université Paris-Est, 61, av. du Général de Gaulle, 94010 Créteil Cedex, France

ARTICLE INFO

Article history:
Received 6 July 2011
Received in revised form 19 March 2012
Accepted 23 May 2012
Communicated by P. Spirakis

Keywords:
Markov chains
Queueing networks
Stochastic bounds

ABSTRACT

We analyze transient and stationary behaviors of multidimensional Markov chains defined on large state spaces. In this paper, we apply stochastic comparisons on partially ordered state which could be very interesting for performance evaluation of computer networks. We propose an algorithm for bounding aggregations in order to derive upper and lower performance measure bounds on a reduced state space. We study different queueing networks with rejection in order to compute blocking probability and end to end mean delay bounds. Parametric aggregation schemes are studied in order to propose an attractive solution: given a performance measure threshold, we vary the parameter values to obtain a trade-off between the accuracy of bounds and the computation complexity.

1. Introduction

Quantitative analysis of multidimensional Markov chains could be very difficult if there is no specific solution form (product-form, matrix geometric solutions, ...). Indeed, we can apply numerical methods, but with the state space explosion problem it is very hard or intractable to compute probability distributions [1]. In performance and dependability studies, the stochastic comparison of Markov chains provides an efficient technique to overcome the state space explosion [11,10]. The key idea of this method is the following: given a large size Markov chain, we propose to bound it by another Markov chain which is easier to analyze, and which provides bounds on performance measures. Various applications of stochastic comparison in queueing networks with many related references are presented in [13]. In [9] stochastic comparison is applied for performance evaluation of different kinds of network architectures. Aggregation techniques have been largely applied for the analysis of large size systems such as queueing networks [2]. In the case of specific networks as closed networks or tandem queueing networks, Norton's theorem is used for aggregating the stations, obtaining an equivalent system with one station. In [1] the Courtois decomposition method is presented for steady state aggregations of large Markov chains. This method can be applied for special systems called decomposable, defined by state subsets that represent tightly coupled structures. In the case of totally ordered state spaces, lumpability and stochastic ordering theory have been combined to derive bounding Markov chains [14].

In the present paper, we use stochastic comparisons by mapping functions [4], in order to aggregate the state space. We can thus reduce the size of large Markov chains by defining bounding Markov chains on a smaller state space. We study stochastic comparisons on state spaces endowed with at least a preorder (so it could be a partial or a total order). The relevance of the paper is to propose algorithms to build aggregated bounding Markov chains. They are based on the

* Corresponding author. Tel.: +33 160764754.
E-mail addresses: hind.castel@it-sudparis.eu (H. Castel-Taleb), lynda.mokdad@univ-paris12.fr (L. Mokdad), nihal.pekergin@univ-paris12.fr (N. Pekergin).

0304-3975/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2012.05.030

definition of many to one mapping functions, and aggregated bounding infinitesimal generators. We can indeed construct the aggregated process without generating the original one, thus there is no supplementary complexity and the bounding procedure can be included in the model construction phase.

We consider the component-wise order which is a partial order on multidimensional state spaces. Note that it is suitable for performance evaluation of communication networks with several nodes. For instance, we can derive bounds for blocking probabilities, mean number of customers and response times in every node. Moreover, using a partial order rather than a total order induces less constraints, since we compare only the transition rates for comparable states. We define parametric aggregations in order to propose a trade-off between reducing the size of the Markov chains, and improving the quality of the bounds. The originality of this paper is to propose a very flexible aggregation scheme allowing to reduce largely the original model. We can start with the smallest bounding model and then gradually increase its size in order to improve the quality of the bounds. This paper is organized as follows. In the next section, we present main theoretical concepts on stochastic comparisons and the stochastic ordering theory. In Section 3, we present the algorithm and its proof. Section 4 presents some applications on different queueing networks for which we compute performance measure bounds using this algorithm. Finally as a conclusion, we present the main contributions of this paper, and we give comments on further research items.

2. Stochastic comparisons

Let A be a discrete and countable state space, and $\preceq$ be at least a preorder relation on A (a reflexive and transitive binary relation [10]). Let us remark that total and partial orders are also preorder relations. For instance, on $A = \mathbb{N}$, $\leq$ is a total order, and on $A = \mathbb{N}^n$, the component-wise order is a partial order, which are both also preorders. We apply the stochastic comparison approach to define a new Markov chain which is an aggregation of the initial one and which provides bounds on the measure of interest. Let $\{X_t, t \geq 0\}$ be a Continuous Time Markov Chain (CTMC) taking values in a large, multidimensional and preordered state space A. We denote by Q the infinitesimal generator. We consider the performability measure $R(t)$ defined as a functional at time t of the considered CTMC:

$$R(t) = \sum_{x \in A} \Pi(x, t) f(x) \tag{1}$$

where $f : A \to \mathbb{R}^+$ is an increasing reward function and $\Pi(x, t)$ is the probability to be in state x at time t. For $t \to \infty$, if the process has a stationary behavior, then we denote by $\Pi(x)$ the stationary probability to be in state x, and R represents the measure of interest computed from the stationary probability distribution Π. If there is no specific solution to compute Π, then its numerical computation is very difficult and may be intractable due to the large state space. Since we are also interested in transient performance measures, we propose to define bounding systems in order to overcome this state space explosion problem. So we reduce A by aggregating some states and by mapping them into one state. We define aggregated processes on the state space $S \subset A$, from which we can derive performance measure $R^u(t)$ such that: $R(t) \preceq R^u(t)$ if it is an upper bound, or $R^l(t)$ such that $R^l(t) \preceq R(t)$ if it is a lower bound. Stochastic comparisons are based on the stochastic ordering theory. Next, we present the main theoretical concepts.

2.1. The stochastic ordering theory

We consider two random variables X and Y defined on A, and their probability measures given respectively by the probability vectors p and q where $p[i] = Prob(X = i), \ \forall i \in A$ (resp. $q[i] = Prob(Y = i), \ \forall i \in A$). The $\preceq_{st}$ ordering can be defined using increasing real functions as follows [10].

Definition 1. $X \preceq_{st} Y \Leftrightarrow E[(f(X))] \leq E[(f(Y))], \forall f : A \to \mathbb{R}^+, \ \preceq -increasing \ whenever \ the \ expectations \ exist$.

We present now the comparison of stochastic processes. Let $\{X_t, t \geq 0\}$ and $\left\{X^u_t, t \geq 0\right\}$ be stochastic processes defined on A.

Definition 2. We say that $\{X_t, t \geq 0\} \preceq_{st} \left\{X^u_t, t \geq 0\right\}$

$$\text{if } X_t \preceq_{st} X^u_t, \forall t \geq 0. \tag{2}$$

When the processes are defined on different state spaces, we can compare them on a common state space using mapping functions. Let X_t (resp. X^u_t) be a process defined on A (resp. S), g be a many to one mapping from A to S. Next, we will compare the mapping of the process X_t by the mapping function g, which means $g(X_t)$ with the process X^u_t. The stochastic comparison by mapping functions is defined as follows [4].

Definition 3. We say that $\{g(X_t), t \geq 0\} \preceq_{st} \left\{X^u_t, t \geq 0\right\}$

$$\text{if } g(X_t) \preceq_{st} X^u_t, \forall t \geq 0. \tag{3}$$

Different formalisms can be used to define a stochastic ordering: increasing functions, and increasing sets [10]. We focus on the increasing set formalism, as we will use it for the proof of our algorithm. Let $\Gamma \subseteq A$, we denote:

$$\Gamma \uparrow = \{y \in A \mid y \succeq x, x \in \Gamma\}. \tag{4}$$

We have the following definition for the increasing set.

Definition 4. Γ is called an increasing set if and only if $\Gamma = \Gamma \uparrow$.

From the general definition of an increasing set, three stochastic orderings have been defined from different families of increasing sets [8]. The most known stochastic ordering $\preceq_{st}$ is derived from the family $\Phi_{st}(A)$ which is defined from all the increasing sets of A:

$$\Phi_{st}(A) = \{\Gamma \subset A \mid \Gamma = \Gamma \uparrow\}. \tag{5}$$

The stochastic ordering $\preceq_{st}$ can be defined between the random variables X and Y using the family of increasing sets $\Phi_{st}(A)$ [8] as follows.

Definition 5. $X \preceq_{st} Y \Leftrightarrow \sum_{x\in\Gamma} p[x] \leq \sum_{x\in\Gamma} q[x], \forall \Gamma \in \Phi_{st}(A)$.

Other families generating weaker orderings are defined from particular increasing sets [8]. Increasing set formalism is widely used for the comparison of multidimensional Markov chains through their infinitesimal generators. It is suitable with algorithms based on matrices. Stochastic comparisons are often based on the monotonicity which is defined as an increase of the process with time t [7].

Definition 6. We say that $\{X_t, t \geq 0\}$ is $\preceq_{st}$-monotone if:

$$X_t \preceq_{st} X_{t+\tau}, \quad \forall t \geq 0, \tau > 0. \tag{6}$$

In [5] it is shown that many queueing systems verify the monotonicity property. As an example, systems represented by multidimensional Markov chains where components are modified by $+1$ (arrivals) or -1 (services), and at the same time -1 in a component and $+1$ in another component (transit between queues) are monotone. These kinds of processes model queueing systems as Jackson or BCMP networks. On the other side, G-Networks are not monotone due to synchronized departures. Generally, the monotonicity of time-homogeneous Markov chains can be proved using the coupling of the chains, by comparing different realizations of the chains with different initial conditions [6,7]. The monotonicity can be also proved by comparing the rows of the generators [8]. If X_t is the Markov chain defined on the state space A with infinitesimal generator Q, then we have the following theorem [8].

Theorem 1. *$\{X_t, t \geq 0\}$ is $\preceq_{st}$- monotone, if $\forall \Gamma \in \Phi_{st}(A)$:*

$$\forall x \preceq y \in A \mid x, y \in \Gamma \text{ or } x, y \notin \Gamma, \quad \sum_{z\in\Gamma} Q[x,z] \leq \sum_{z\in\Gamma} Q[y,z].$$

We consider now X_t as defined previously and also the Markov chain X^u_t defined on another state space S with infinitesimal generator Q^u, and g a many to one mapping from A to S. If we suppose that at least one process is monotone, then the theorem of the comparison of these Markov chains by the mapping function g is as follows [8].

Theorem 2. *If the following conditions 1, 2, 3 are satisfied:*

1. $g(X_0) \preceq_{st} X^u_0$
2. *$\{X_t, t \geq 0\}$ or $\{X^u_t, t \geq 0\}$ is $\preceq_{st}$-monotone*
3. $Q[x,*]M_g \preceq_{st} Q^u[y,*], \forall x \in A, y \in S,\ g(x) = y$

then we have:

$$\{g(X_t), t \geq 0\} \preceq_{st} \left\{X^u_t, t \geq 0\right\} \tag{7}$$

where $Q[x,*]$ is the row in the matrix Q corresponding to the state x, and M_g, is the matrix representation of the mapping function g described as follows [8]:

$$M_g[i,j] = \begin{cases} 1 & \text{if } g(i) = j \\ 0 & \text{else} \end{cases}, \qquad i \in A \text{ and } j \in S.$$

The following relation between the rows of the generators:

$$Q[x,*]M_g \preceq_{st} Q^u[y,*]$$

is equivalent to:

$$\sum_{g(z)\in\Gamma} Q(x,z) \leq \sum_{z\in\Gamma} Q^u(y,z), \quad \forall \Gamma \in \Phi_{st}(S). \tag{8}$$

This theorem can be also used to compare the Markov chain $\{X_t, t \geq 0\}$ with a lower bound $\{X^l_t, t \geq 0\}$, by reversing the inequalities, and by considering the generator Q^l instead of Q^u. We deduce that Theorem 2 is based on three conditions in order to have the comparison by mapping functions of Markov chains defined on different state spaces : (1) the comparison of the chains at initial time, (2) the monotonicity of at least one chain, and (3) the comparison of the generators. Next, we use these theoretical concepts in order to define algorithms generating aggregated bounding chains.

3. Bounding aggregations

We use the theoretical concepts of stochastic comparisons by mapping functions in order to propose an algorithmic approach to generate aggregated bounding chains for the computation of upper or lower bounds on performance measures.

3.1. Presentation of the algorithm

The key idea of Algorithm 1 is to build two aggregated bounding Markov chains: an upper bound, $X^u(t)$ and a lower bound, $X^l(t)$ on $g(X_t)$, at each time t. For ease of presentation, we give the algorithm only for the upper bound. In Algorithm 1, first we have to define in (1) a preorder in the state space such that the reward function in an increasing function. Second, in (2) we verify that the CTMC to aggregate is monotone (as it is a condition of Theorem 2). After that, we have two main steps for the definition of the aggregated bounding Markov chains: first we define the many to one mapping function g in order to reduce the state space A; second we define the infinitesimal generators Q^u of the aggregated upper bounding CTMC.

3.1.1. Definition of the many to one mapping function

The many to one mapping function $g : A \to S$ ($S \subset A$) is defined by aggregating some states of A. We have two cases for a state $x_i \in A$.

1. The state $x_i \in A$ is not aggregated with other states, which means that x_i is mapped to the same state x_i, so it is represented exactly and it is called a simple state of S. So we have: $g(x_i) = x_i$, and $\nexists y \in A \mid y \neq x_i$ *and* $g(y) = x_i$.
2. The set of states $\{x_1, \ldots, x_i, \ldots, x_n\}$, is mapped to x_n which is the upper state ($\forall 1 \leq i \leq n, x_i \preceq x_n$), and x_n is a macro-state. Note that in the case of the lower bound we map to the lower state x_1 which is a macro-state.

So g is such that $g(x_1) = g(x_2) = \cdots = g(x_i) = \ldots g(x_n) = x_n$ for the upper bound. Obviously, the macro-states represent disjoint sets of states. It is also important to note that g is an increasing function since we compute bounds defined as an increasing reward function on a probability distribution of the CTMC. From definition of g, we obtain the process $\{g(X_t), t \geq 0\}$, where states correspond to the mapping $g(X_t)$ of X_t.

3.1.2. Building the aggregated bounding Markov chain

We define the infinitesimal generator Q^u from Q and M_g defined previously. The product QM_g is the matrix representing the transition rates from states $x \in A$ into states $y \in S$. The infinitesimal generator Q^u is defined as follows:

$$\forall x \in S, \quad Q^u[x, *] = Q[x, *]M_g \tag{9}$$

where $Q^u[x, *]$ is the row in the matrix Q^u corresponding to the state x. Note that if we define a lower bound, then after the definition of the mapping function g as the mapping to the lower state, then we deduce the generator Q^l as in Eq. (9).

Algorithm 1 Construction of an aggregated upper bounding Markov chain X_t^u such that $R(t) \leq R^u(t)$

Require: infinitesimal generator Q, state space A, reward function f used to define the underlying performance measure $R(t)$.

1: Define a preorder $\preceq$ on A such that f is an increasing function with respect to this order
2: Check the monotonicity of $\{X_t, t \geq 0\}$
3: For each state $x \in A$, give $y \in S$ such that $y = g(x)$. Note that y can be a simple or a macro-state. If it is a macro-state, then y is the upper state of the set.
4: For each state $x \in S$ define the row in Q^u corresponding to the transition rates from state x : $Q^u[x, *] = Q[x, *]M_g$.
5: If $\{X_t^u, t \geq 0\}$ is not irreducible, then return to (3), otherwise, go to the next step.
6: Compute $\Pi^u(t)$, and $R^u(t)$.

In step (4) of Algorithm 1, the bounding generator is constructed by multiplying the generator Q with M_g. Since the entries of M_g are 0 or 1, the product QM_g is computed by additions of entries of Q. Thus for each state in A the number of additions is bounded by the size of S. The complexity of step (4) is thus in $O(|A| \times |S|)$, where $|A|$ and $|S|$ represent respectively the sizes of the state spaces A and S. Once the bounding chain is defined, it is numerically analyzed by conventional techniques [12]. Note that step (2) can be concluded directly since for some classes of systems their monotonicity is established. For the definition of the many to one mapping function, the choice of states to aggregate is not simple. Different criteria are considered: the irreducibility of the aggregated Markov chain, and the quality of the bounds. In model-driven performance studies, this procedure can be done with heuristics: by taking into account the dynamic of the original model, the bounding model can be derived by modifying some parameters and/or characteristics of the model. Thus the bounding model can be directly constructed without storing the generator Q. In the case the original model is not monotone, we can generalize this algorithm by defining monotone bounding models, since for stochastic comparisons, the monotonicity of one of the models is sufficient. Next, we prove that the aggregated Markov chain provides really bounds.

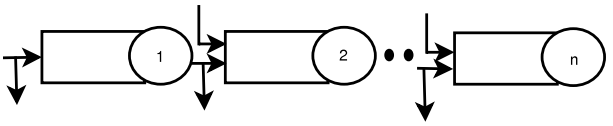

Fig. 1. A tandem queueing network.

3.2. Proofs

We give the proof only for the upper bound case. From Algorithm 1, we define the mapping function $g : A \to S$ such that the states are aggregated and mapped to the upper state. We apply Theorem 2 in order to prove that:

$$\{g(X_t), t \geq 0\} \preceq_{st} \{X_t^u, t \geq 0\}. \tag{10}$$

We have two cases for the mapping function g : either it maps a state to itself, or it maps some states into the upper state (aggregation). In the first case, as there is only one state $x \in A$ such that $g(x) = x$, and as $Q^u[x, *] = Q[x, *]M_g$, it is easy to deduce that:

$$\forall x \in A, y \in S \mid g(x) = y, \quad Q[x, *]M_g \preceq_{st} Q^u[y, *]. \tag{11}$$

In the case where the states $x_1, \ldots, x_n \in A$ such that $x_i \preceq x_n, \forall i \leq n$ are mapped to x_n, then from the monotonicity property of $\{X_t, t \geq 0\}$ [8] we have:

$$Q[x_1, *] \preceq_{st} Q[x_n, *] \tag{12}$$

$$Q[x_2, *] \preceq_{st} Q[x_n, *] \tag{13}$$

$$\ldots$$

$$Q[x_{n-1}, *] \preceq_{st} Q[x_n, *]. \tag{14}$$

Moreover, since g is an increasing function, then $\forall\ 1 \leq i \leq n$ we have:

$$Q[x_i, *]M_g \preceq_{st} Q[x_n, *]M_g. \tag{15}$$

As $Q^u[x_n, *] = Q[x_n, *]M_g$, then it is easy to deduce that:

$$\forall x \in A, y \in S \mid g(x) = y, \quad Q[x, *]M_g \preceq_{st} Q^u[y, *]. \tag{16}$$

From Theorem 2, if condition (1) is verified then we deduce that:

$$\{g(X_t), t \geq 0\} \preceq_{st} \left\{X_t^u, t \geq 0\right\}. \tag{17}$$

From the stochastic comparison of Markov chains by mapping functions, we obtain the comparison of transient and stationary probability distributions. So we can deduce that for the upper bound:

$$R(t) \leq R^u(t) = \sum_{x \in S} \Pi^u(x, t) f(x). \tag{18}$$

For the lower bound, the algorithm considers the mapping function $g : A \to S$ to the lower state of the set, and we derive M_g. We can compute Q^l for each state $x \in S$: $Q^l[x, *] = Q[x, *]M_g$. From Theorem 2, we deduce that the aggregated process $\{X^l(t), t \geq 0\}$ allows to compute a lower bound $R^l(t)$:

$$R^l(t) = \sum_{x \in S} \Pi^l(x, t) f(x) \leq R(t). \tag{19}$$

Next, we give some numerical results obtained from the algorithm for computation of performance measure bounds using bounding aggregations.

4. Applications

We present in this section some applications on different queueing networks to illustrate how different performance measure bounds can be obtained from bounding aggregations. First, we apply our algorithm on a tandem queueing network, and after on a more general queueing network with feedback.

4.1. A tandem queueing network

The first considered model is an open tandem queueing network which may represent a path in a network defined as a series of network nodes (switches, routers). We suppose that the leftmost node has index 1, and indices increase in the path until node n (see Fig. 1).

For any queue i such that $1 \leq i \leq n$, arrivals follow a Poisson process with rate λ_i, the service time is exponential with rate μ_i, and the capacity is B_i. After a service in queue i, the packet goes out with probability d_i or transits to the next queue $i + 1$ with probability $p_{i,i+1}$ if $i < n$. In the case when the queue is full, the customer is lost. This system can be represented by a CTMC $\{X_t,\ t \geq 0\}$ on $A = \{0, \ldots, B_1\} \times \cdots \times \{0, \ldots, B_i\} \times \cdots \times \{0, \ldots, B_n\}$. Each state $x \in A$ is represented by the vector $x = (x_1, \ldots, x_i, \ldots, x_n)$, where x_i is the number of customers (or packets) waiting in queue i.

4.1.1. Performance measures

We explain how to compute the blocking probabilities at each node i and the end to end delays. The blocking probability $Pb_i(t)$ of queue i at time t is given by:

$$Pb_i(t) = \sum_{x \in A | x_i = B_i} \Pi(x, t). \tag{20}$$

The mean number of packets at queue i at time t is given by $N_i(t)$:

$$N_i(t) = \sum_{x \in A} x_i \Pi(x, t). \tag{21}$$

In the case where the stationary distribution Π exists, we denote limiting cases of the number of packets and the blocking probabilities by N_i and Pb_i. The end to end mean delay can be computed by means of the Little formula. Let Λ_i be the throughput, D_i the mean delay and N_i the mean number of packets at queue i. It follows from the Little formula:

$$D_i = \frac{N_i}{\Lambda_i}. \tag{22}$$

Since in the equilibrium, the throughput of a queue is equal to the arrival rate of accepted packets, it can be computed as follows: $\Lambda_1 = \lambda_1(1 - Pb_1)$, and $\Lambda_{i+1} = (\Lambda_i p_{i,i+1} + \lambda_{i+1})(1 - Pb_{i+1})$ for $0 \leq i \leq n-1$. The end to end mean delay D is then obtained by summing the mean delays D_i through the path:

$$D = \sum_{i=1}^{n} D_i. \tag{23}$$

The computation of the stationary distribution for $\{X_t,\ t \geq 0\}$ is very difficult or intractable: there is no product-form, and the number of states increases exponentially with the number of components. In [3], performance analysis of tandem queueing networks has been proposed using stochastic bounds. Their results will be compared with those obtained by our algorithm.

4.1.2. Application of the algorithms

We propose to apply Algorithm 1 for the generation of aggregated bounding processes in order to derive performance measure bounds. Two conditions must be satisfied to apply the algorithm. The first one is the definition on the state space A of an order such that we can compute bounds for Pb_i and D. We propose the component-wise partial order:

$$\forall x,\ y \in A,\ x \preceq y \Leftrightarrow x_1 \leq y_1, \ldots, x_n \leq y_n.$$

We can see from Eqs. (21) and (20) that according to the component-wise ordering, these performance measures are defined as increasing rewards on the probability distributions. Thus if $\Pi(t) \leq_{st} \Pi^u(t)$ then the mean number of packets and the blocking probability computed from $\Pi(t)$ are less than that of computed from $\Pi^u(t)$. For the average delay in the equilibrium, we can see that in order to compute an upper bound on D, we must consider an upper bound on N_i and a lower bound on Λ_i. Thus the upper bound on D can be derived by computing N_i^u and Pb_i^u from the upper bounding stationary distribution Π^u. Similarly the lower bound on D can be computed from the lower bounding stationary distribution. The second condition is the monotonicity of $\{X_t, t \geq 0\}$ which has been proved in [5] using the coupling by event. Thus, the aggregated processes $\{X_t^u, t \geq 0\}$ and $\{X_t^l, t \geq 0\}$ can be derived through the algorithm generating bounds for the performance measures.

For the definition of the mapping function g, we propose parametric aggregation schemes based on two parameters: Δ and k. Let k be a number of queues (such as $k < n$), and $\Delta \leq B_i, \forall 1 \leq i \leq n$. The general idea of the aggregation scheme is the following: for a state $x = (x_1 \ldots, x_k, \ldots, x_n) \in A$, we change values of components $x_1, \ldots, x_k$, (corresponding to queues $1, \ldots, k$) to upper or lower values, and we keep unchanged the other components. Obviously, the value of k will have an impact on the quality of the bounds: if k increases the bounds will be degraded. Moreover, for all the aggregation schemes, the component x_n has not been changed, as we compute the blocking probability of queue n. Next, we describe the upper bound defined by overestimating the number of packets, and the lower bound by underestimating the number of packets in the modified k queues, as follows.

1. For the upper bound: states x such that $x_i \leq B_i - \Delta, \forall 1 \leq i \leq k$, are mapped to $(B_1 - \Delta, \ldots, B_k - \Delta, x_{k+1}, \ldots, x_n)$.
2. For the lower bound: states x such that $x_i \geq \Delta, \forall 1 \leq i \leq k$, are mapped to the lower state $(\Delta, \ldots, \Delta, x_{k+1}, \ldots, x_n)$.

For $\Delta = 0$ we can see that the size of the state space S is $(B+1)^{n-k}$. Hence the state space size is largely reduced when k increases compared with the size of the exact process which is $(B+1)^n$. Note that we could also define bounding systems with different values of Δ for each queue. The main advantage of these aggregation schemes is the possibility to obtain a very reduced Markov chain, and to increase gradually the size in order to improve the quality of the bounds.

Table 1
The blocking probabilities computed from the $M/M/1/B_i$ queue.

λ	20	30	40	50	60	70	80	90
Pb_4^{u*}	2.6e−5	0.1667	0.375	0.5122	0.5811	0.6413	0.6822	0.7223

Table 2
Blocking probability bounds in queue 4: impact of k.

λ	Pb_4^u $(k=3)$ size = 41	Pb_4^u $(k=2)$ size = 1681	Pb_4^u $(k=1)$ size = 68921	Pb_4^l $(k=3)$ size = 41	Pb_4^l $(k=2)$ size = 1681	Pb_4^l $(k=1)$ size = 68921
20	1.5e−3	8.6e−5	1.0e−5	0	1.0e−16	1.0e−14
30	0.02439	0.01628	0.01401	0	9.0e−13	1.7e−8
40	0.0927	0.0917	0.0916	0	6.3e−8	6.2e−4
50	0.1667	0.1666	0.1666	4.0e−13	2.2e−4	8.8e−2
60	0.2307	0.2307	0.2307	5.0e−10	0.0352	0.1811
70	0.2857	0.2857	0.2857	1.0e−7	0.1597	0.2856
80	0.3333	0.3333	0.3333	2.0e−5	0.2646	0.3011
90	0.3751	0.3751	0.3751	1.0e−3	0.3460	0.3522

Table 3
End to end delay bounds: impact of Δ.

λ	D size = 2825761	D^u $(\Delta=5)$ size = 8856	D^u $(\Delta=20)$ size = 379701	D^l $(\Delta=20)$ size = 379701
20	0.066	1.190	0.824	0.036
30	0.105	1.309	1.009	0.075
40	0.348	1.420	1.181	0.299
50	0.742	1.469	1.269	0.538
60	0.988	1.493	1.316	0.653
70	1.131	1.508	1.348	0.726
80	1.182	1.520	1.378	0.770
90	1.241	1.529	1.414	0.811

4.1.3. Numerical results

Next, we consider an example of network with n=4 queues in tandem. For each queue $1 \leq i \leq 4$, the arrival rate λ_i is such that $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_2 = \lambda$. We have also the following assumptions: $\mu_i = 100$ Mb/s, and $B_1 = \cdots = B_n = B$. The routing probabilities are such that: $\forall 1 \leq i \leq n$, $d_i = 0.3, \forall 1 \leq i \leq n-1$, $p_{i,i+1} = 0.7$.

We take a buffer size B equal to 40 for each queue. In Table 2, we give the steady state blocking probability bounds of queue 4, for different values of k and for $\Delta = 0$. In order to show the interest of our results, we compare our upper bounds with those obtained in [3]. In this paper, the blocking probability upper bounds of any queue i $(1 \leq i \leq n)$ in a tandem queueing network of n queues are derived from the blocking probability of an $M/M/1/B_i$ queue, by supposing that all previous queues $\{1, \ldots, i-1\}$ have an infinite capacity. We denote by Pb_4^{u*} the blocking probability for queue 4, and we give in Table 1 the values obtained using the same input parameter values as those used for Table 2 in order to compare them.

First, we remark in Table 1 that values obtained from Pb_4^{u*} are larger than upper bounds of Table 2. If we study in detail Table 2, then we see that when the number of modified queues, k decreases, the quality of bounds is improved. Furthermore, when λ is greater than 50, k has no impact on the upper bounds because when λ is sufficiently large, then all queues are full for the upper bound. For the lower bound, the last $n-k$ queues are full, and arrivals from the k queues are neglected. For this reason we can see that in the case of $k=3$, the lower bounds are very low (lower than 10^{-16} for $\lambda \leq 40$).

From Table 2, we can use parameter k in order to verify step by step that Pb_4 is in an interval. For example, for $\lambda = 60$, if we want to check that the blocking probability is in the interval [0.1, 0.3], we begin with the smallest CTMC: $k=3$ for the upper bound (size of 41) and $k=2$ (size of 1681) for the lower bound. We obtain the interval [0.0352, 0.2307] of bounding values. Since we need a more accurate lower bound, we decrease k. For $k=1$ (size of 68921), we obtain the lower value equal to 0.1811. So with the interval [0.1811, 0.2307] we have verified the constraint on blocking probabilities, without computing the exact Markov chain of size 2825761 states. In Table 3, we give the exact end to end delays, and also the upper and lower bounds D^u and D^l. We take $B=40$, $k=3$, and we consider different values of Δ.

We can notice in Table 3 the influence of Δ, if it increases the bounds are more accurate. Next, we take a larger system with 9 queues. We suppose that $B=20$, $n=9$, $k=5$ and $\Delta=1$. In Table 4, we give the blocking probabilities and end to end mean delay bounds. We can see that the size of the exact CTMC is 21^9, which is intractable. The sizes of bounding models are reduced to $21^4 \times 2^5$ and we can have estimations to check if constraints on performance measures are satisfied or not. From these results, we have noticed the influence of both parameters Δ and k on accuracy of bounds. First, we choose

Table 4
Blocking probabilities and end to end mean delay bounds.

λ	Pb_9^u	Pb_9^l
20	5.0e−4	6.1e−7
30	2.6e−2	9.8e−4
40	0.097	0.051
50	0.168	0.131
60	0.231	0.211
70	0.285	0.261
80	0.333	0.311
90	0.375	0.361
λ	D^u	D^l
20	1.376	0.066
30	1.376	0.104
40	1.531	0.220
50	1.620	0.365
60	1.670	0.469
70	1.700	0.537
80	1.720	0.580
90	1.733	0.619

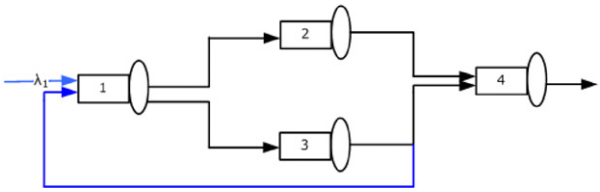

Fig. 2. A queueing network with feedback.

Table 5
Blocking probability bounds in a queueing network with feedback.

λ	Pb_4 size = 194481	$Pb_4^u(\Delta = 5)$ size = 15876	$Pb_4^u(\Delta = 10)$ size = 53361	$Pb_4^l(\Delta = 10)$ size = 53361	$Pb_4^l(\Delta = 15)$ size = 112896
20	2.0e−13	1.7e−3	2.2e−6	6.5e−15	8.2e−15
30	5.7e−10	6.8e−3	4.6e−5	1.7e−11	2.3e−11
40	1.5e−7	1.8e−2	4.6e−4	3.9e−9	6.0e−9
50	9.8e−6	3.8e−2	2.7e−3	2.3e−7	4.0e−7
60	2.6e−4	6.6e−2	1.1e−2	5.1e−6	1.0e−5
70	0.004	0.103	0.03	5.4e−5	1.2e−4
80	0.023	0.145	0.066	3.0e−4	7.0e−4
90	0.074	0.18	0.114	9.0e−4	2.0e−3

the value of k as it has an important impact on the size of the state space, and after that we choose the value of Δ in order to improve the quality of the bounds.

4.2. A queueing network with feedback

We study a more general queueing network with feedback given in Fig. 2. We suppose that arrivals to the system are into queue 1 and follow a Poisson process. For each queue i ($1 \leq i \leq 4$), the service rates are exponential with parameter $\mu_i = 100$. The routing probabilities between the queues are as follows: $p_{1,2} = p_{1,3} = 0.5$, $p_{3,1} = 0.3$, and $p_{3,4} = 0.7$. We suppose that $B_1 = B_2 = B_3 = B_4 = 20$, and $k = 2$. The system can be represented by a multidimensional CTMC which is monotone. The monotonicity of Jackson networks has been proved in [7], and considering finite capacities does not modify the monotonicity.

In Table 5, we give the blocking probabilities (exact and bounds) of queue 4, for different values of Δ.

In Table 6, we give the transient blocking probabilities (exact and bounds) for different values of t. We choose the initial probability vector such that the probability is one in the greatest state (when all the queues are full) and 0 in others.

We can remark in Table 6 that transient bounds are decreasing in time. This is due to the monotonicity: the original model is monotone and since the many to one application, g is increasing, the bounding model is also monotone. Since the

Table 6
Transient blocking probability bounds: the monotonicity behavior.

t	$Pb_4(t)$ size = 194481	$Pb_4^u(t)$ ($\Delta = 10$) size = 53361	$Pb_4^l(t)$ ($\Delta = 10$) size = 53361
$1.0e{-}5$	0.9991	0.9999	0.998
$1.0e{-}3$	0.9137	0.9138	0.9120
$1.0e{-}2$	0.6655	0.666	0.6211

initial state is the greatest state, the convergence to the stationary distribution will be decreasing over time. From numerical results presented in this section, we can conclude that by varying the aggregation parameter we can have a trade-off between accuracy of bounds, and computation complexity.

5. Conclusion

We apply bounding aggregations on Markov processes in order to compute upper and lower performance measure bounds. Different queueing networks are proposed in order to show the relevance of the approach. Moreover, as we have applied stochastic comparisons, then both stationary and transient bounds are derived. We develop a parametric aggregation scheme which allows to have a trade-off between accuracy of bounds and computation complexity. As a future work, we need to define an algorithm for nonmonotone Markov processes by building monotone bounding processes. It will be also interesting to generalize this algorithm to weaker stochastic orderings.

References

[1] G. Bolch, S. Greiner, H. Meer, K.S. Trivedi, Queueing Networks and Markov Chains, Modelling and Performance Evaluation with Computer Science Applications, Second Edition, Wiley, 2006.
[2] H. Chen, D. Yao, Fundamentals of queueing networks, in: Application of Mathematics, Stochastic Modelling and Applied Probability, Springer, 2001.
[3] Y. Dallery, L. Truffet, Bounds on the End to End loss probability for queues in series with finite capacity, Stochastic models 16 (5) (2000) 557–563.
[4] M. Doisy, A coupling technique for stochastic comparison of functions of Markov processes, Journal of Applied Mathematics and Decision Sciences 4 (1) (2000) 39–64.
[5] P. Glasserman, D.D. Yao, Monotone structure in discret event systems, in: Wiley Series in Probability and Mathematical Statistics-applied Probability and Statistics Section, 1994.
[6] T. Lindvall, Lectures on the Coupling Method, Wiley series in Probability and Mathematical Statistics, 1992.
[7] T. Lindvall, Stochastic monotonicities in Jackson queueing networks, Probability in the Engineering and Informational Sciences 11 (1997) 1–9.
[8] W. Massey, Stochastic orderings for Markov processes on partially ordered spaces, Mathematics of Operations Research 12 (2) (1987).
[9] L. Mokdad, H. Castel-Taleb, Stochastic comparisons: a methodology for the performance evaluation of fixed and mobile networks, Computer Communications 31 (17) (2008).
[10] A. Muller, D. Stoyan, Comparison methods for Stochastic Models and Risks, in: J. Wiley and son in Probability and Statistics, 2002.
[11] M. Shaked, J. Shantikumar, Stochatic Orders and their Applications, Academic Press, Boston, 1994.
[12] W.J. Stewart, Introduction to the numerical solution of Markov chains, Princeton University Press, 1994.
[13] R. Szekli, Stochastic orderings for queueing networks, http://www.math.uni.wroc.pl/szekli/documents/networks2007.pdf.
[14] L. Truffet, Reduction technique for discrete time Markov chains on totally ordered space using stochastic comparisons, Journal of Applied Probability 37 (3) (2000).