Stochastic Bounds and Stochastic Monotonicity: methods, algorithms and applications

J.M. Fourneau[1] and N. Pekergin[2]

Laboratoire PRiSM, CNRS UMR 8144

Université de Versailles St-Quentin

[1] ***INRIA Mescal***

[2] ***Université Val de Marne Paris 12***

[a] Joint works with Mouad Ben Mamoun (Mohammed V University, Rabat), Ana Bušić, Tugrul Dayar (Bilkent University), and Jean-Marc Vincent (LIG-CNRS-INRIA Mescal).

Outline

- Motivation and some questions
- The classical framework for Markov Chains: strong stochastic ordering, total ordering of the state space, steady-state analysis of DTMC
- Getting more from the classical framework
- Beyond the classical framework: partial ordering of the states, variability ordering among random variables

Motivation

- Solving very large Markov chains.
- Solving a set of chains (worst case analysis).
- Qualitative properties of models based on Markov chains.
- Proof of algorithms based on Markov chains.

Solving Large Chains

- The composition of submodels in interaction allows modeling of large and complex systems.
- A tensor representation of MC, either in discrete-time or continuous-time [30, 43]:

$$P = \sum_i \otimes_j M_i^j.$$

- Associated to several High Level Formalisms (Stochastic Process Algebra, Stochastic Automata Networks, Superposition of Stochastic Petri Nets, etc..).
- An efficient storage of large chains.

- But numerical analysis of chains in steady-state is still difficult [43].
- Compute performance indices R defined as reward functions on the steady-state distribution:

$$R = \sum_i r(i)\pi(i).$$

- In general the tensor representation is less efficient than the usual sparse matrix form for basic operations required for numerical analysis.

Bounding the Rewards

- Exact values of the performance indices are sometimes not necessary.
- It is often sufficient to satisfy the Quality of Service (QoS) requirements.
- Bounding some reward functions is sufficient.

Bounds

- Linear algebra problem ($\pi = \pi P$), polyhedral properties (Courtois and Semal [17, 18], Goyal, Muntz, Lui, Rubino and Buchholz [8]).
- Markov Decision Process (Van Dijk [49]).
- Stochastic Bounds (bounds of the sample-paths, coupling) (Stoyan [44, 45], Kijima [32], Shaked, Shantikumar[42]).
- Here : stochastic comparison and stochastic monotonicity based on linear algebra, not on sample-path theorem or coupling (stochastic arguments).

Methodology

- We have to model a problem using a very large Markov chain and compute its steady-state distribution.
- Design algorithmically a new chain (transition matrix) such that:
  - The reward functions will be upper or lower bounds of the exact reward functions.
  - The new matrix is simpler to solve (smaller or with an easy structure).
- Based on stochastic ordering and monotonicity of Markov chains, lumpability (Truffet) or censoring (Younès) for building smaller chains) and patterns (Busic) for the derivation of structured DTMC.

Motivation again: worst case analysis

- Models where some parameters are not perfectly known.
- For instance: transition probabilities are in some interval.
- Solving the worst case in the set of DTMC (i.e. the worst average reward).
- How to find the "worst" matrix in a set ?
- For steady-state and transient rewards, and absorption time or probabilities.
- Based on stochastic orderings for random variables and Markov chains, monotonicity of DTMC.

Motivation continued: Qualitative Properties

- Prove that a steady-state or transient reward or an absorbing time is increasing with a parameter or the DTMC.
- Prove the convergence of algorithms based on a Markov chain.
- Based on the monotonicity of the DTMC.

The classical methodology and framework

- Total ordering of the states.
- Strong stochastic ordering of the chain.
- Steady-state analysis.

Strong Stochastic Bounds

- Restriction (here) : Discrete Time Markov Chains (DTMC) with **finite state space** $E = \{1, \ldots, n\}$ (n is the size of the chain) and **total order on the state space**.
- Continuous-Time MC : will be studied in the next section
- $P_{i,*}$ will refer to row i of P.

Comparison of Random Variables

- The strong stochastic ordering is defined by the set of non-decreasing functions or by matrix K_{st} (Stoyan [44]).
- **Definition 1** *Let X and Y be random variables taking values on a totally ordered space. Then $X <_{st} Y$ if and only if $E[f(X)] \leq E[f(Y)]$ for all non decreasing functions f whenever the expectations exist.*

Discrete states

- **Definition 2** *If X and Y take values on the finite state space $\{1, 2, \ldots, n\}$ with p and q as probability distribution vectors, then $X <_{st} Y$ if and only if $\sum_{j=k}^{n} p_j \leq \sum_{j=k}^{n} q_j$ for $k = 1, 2, \ldots, n$, or briefly:*

  $pK_{st} \leq qK_{st}$ component-wise (i.e. $pK_{st} \leq_{el} qK_{st}$).

- $K_{st} = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 1 & 0 & \ldots & 0 \\ 1 & 1 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \ldots & 1 \end{bmatrix}$

Example

$$(0.1, 0.3, 0.2, 0.1, 0.3) <_{st} (0, 0.4, 0, 0.3, 0.3)$$

because

$$\begin{cases} 0.3 & \leq & 0.3 \\ 0.1 + 0.3 & \leq & 0.3 + 0.3 \\ 0.2 + 0.1 + 0.3 & \leq & 0 + 0.3 + 0.3 \\ 0.3 + 0.2 + 0.1 + 0.3 & \leq & 0.4 + 0 + 0.3 + 0.3 \\ 0.1 + 0.3 + 0.2 + 0.1 + 0.3 & \leq & 0 + 0.4 + 0 + 0.3 + 0.3 \end{cases}$$

Example

- $x = (0.1, 0.3, 0.2, 0.1, 0.3)$ and $y = (0, 0.5, 0, 0.2, 0.3)$ are not st-comparable because:
- $0.1 + 0.3 \leq 0.2 + 0.3$; thus $y <_{st} x$ is not true.
- $0.2 + 0.1 + 0.3 \geq 0 + 0.2 + 0.3$; thus $x <_{st} y$ is not true.

st-bounds

- Average population, loss rates or tail probabilities are non decreasing functions.
- Bounds on the distribution imply bounds on these performance indices as well.
- St-bounds are valid for transient distributions as well as the steady state (we first study the steady-state here).

Comparison for Markov Chains

- Monotonicity [31] and comparability of the transition probability matrices yield sufficient conditions for the stochastic comparison of MC.
- **Definition 3 (st-Comparison of Stochastic Matrices)** *Let P and Q be two stochastic matrices. $P <_{st} Q$ if and only if $PK_{st} \leq QK_{st}$. This can be also characterized as $P_{i,*} <_{st} Q_{i,*}$ for all i.*

st-Monotone Matrix

- **Definition 4 (St-Monotone Matrix)** *Let P be a stochastic matrix, P is st-monotone if and only if for all u and v, if $u <_{st} v$ then $uP <_{st} vP$.*
- St-monotone matrices are completely characterized (this is not true for other orderings, see [5]).
- **Definition 5** *Let P be a stochastic matrix. P is st-monotone if and only if $K_{st}^{-1} P K_{st} \geq 0$ component-wise.*
- **Property 1** *Let P be a stochastic matrix, P is st-monotone if and only if for all i, $j > i$, we have $P_{i,*} <_{st} P_{j,*}$*

Examples

- $\begin{bmatrix} 0.1 & 0.2 & 0.6 & 0.1 \\ 0.1 & 0.1 & 0.2 & 0.6 \\ 0.0 & 0.1 & 0.3 & 0.6 \\ 0.0 & 0.0 & 0.1 & 0.9 \end{bmatrix}$ is monotone.
- $\begin{bmatrix} 0.1 & 0.2 & 0.6 & 0.1 \\ 0.2 & 0.1 & 0.1 & 0.6 \\ 0.0 & 0.1 & 0.3 & 0.6 \\ 0.1 & 0.0 & 0.1 & 0.8 \end{bmatrix}$ is not monotone.

Fundamental theorem

Theorem 1 *Let $X(t)$ and $Y(t)$ be two DTMC and P and Q be their respective stochastic matrices. If*

- $X(0) <_{st} Y(0)$,
- *st-monotonicity of at least one of the matrices holds,*
- *st-comparability of the matrices holds, that is, $P_{i,*} <_{st} Q_{i,*} \;\; \forall i$.*

Then $X(t) <_{st} Y(t), t > 0$.

Proof

- By induction on t:
- Assume that $X(t) <_{st} Y(t)$ (true for $t = 0$).
- Then, $X(t)P <_{st} X(t)Q$ (simple lemma).
- Assume Q is st-monotone,
  as $X(t) <_{st} Y(t)$ we have: $X(t)Q <_{st} Y(t)Q$.
- Thus, $X(t)P <_{st} Y(t)Q$.
- After identification $X(t+1) <_{st} Y(t+1)$.

Relations

- Thus, assuming that P is not monotone, we obtain a set of inequalities on the elements of Q:

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

Algorithms

- It is possible to use a set of equalities, instead of inequalities:

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

- Properly ordered (in increasing order for i and in decreasing order for j in previous system), a constructive way to obtain a stochastic bound (Vincent's algorithm [1]).

Vincent's Algorithm

Construction of an upper bound Q *:*$P <_{st} Q$ *and* Q *is* $<_{st}$ *monotone*

Column n*:*

$Q_{1,n} = P_{1,n}$;

For $i = 2$ **to** n **Do** $Q_{i,n} = \max(P_{i,n}, Q_{i-1,n})$;

Column $j,\ n-1 \geq j \geq 2$*:*

For $j = n-1$ **downto** 2 **Do**

$Q_{1,j} = P_{1,j}$;

For $i = 2$ **to** n **Do**

$Q_{i,j} = \max(\sum_{k=j}^{n} P_{i,k}, \sum_{k=j}^{n} Q_{i-1,k}) - \sum_{k=j+1}^{n} Q_{i,k}$;

End

End

Column 1*:*

For $i = 1$ **to** n **Do** $Q_{i,1} = 1 - \sum_{k=2}^{n} Q_{i,k}$;

Technical details

- For the sake of simplicity, we use a full matrix representation for P and Q.
- Stochastic matrices for real problems are usually sparse.
- The sparse matrix and tensor versions of most of the algorithms are straightforward.
- **Definition 6** *We denote by* $v(P)$ *the matrix obtained after application of Vincent's Algorithm to a stochastic matrix* P*.*

An example

$$P1 = \begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.2 & 0.0 \\ 0.1 & 0.7 & 0.1 & 0.0 & 0.1 \\ 0.2 & 0.1 & 0.5 & 0.2 & 0.0 \\ 0.1 & 0.0 & 0.1 & 0.7 & 0.1 \\ 0.0 & 0.2 & 0.2 & 0.1 & 0.5 \end{bmatrix}$$

- Once an element is obtained, we can compute the element on the left and below.
- Begin with element $(1, n)$.
- Proceed by row or by column.
- The summations $\sum_{k=j}^{n} Q_{i-1,j}$ and $\sum_{k=j+1}^{n} Q_{i,j}$ are already computed when we need them. Store to avoid computations.

First steps

- First row is unchanged:

$$\begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.2 & 0.0 \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix}.$$

First column

- Compute column n (st-monotonicity implies that the elements are non decreasing):

$$\begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.2 & 0.0 \\ & & & & 0.1 \\ & & & & 0.1 \\ & & & & 0.1 \\ & & & & 0.5 \end{bmatrix}.$$

Next Column

- Compute column $n-1$ (st-monotonicity implies that the sums of the last two elements in a row are non decreasing):

$$\begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.2 & 0.0 \\ & & & 0.1 & 0.1 \\ & & & 0.1 & 0.1 \\ & & & 0.7 & 0.1 \\ & & & 0.3 & 0.5 \end{bmatrix}.$$

- Finally $Q = v(P1) = \begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.2 & 0.0 \\ 0.1 & 0.6 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.5 & 0.1 & 0.1 \\ 0.1 & 0.0 & 0.1 & 0.7 & 0.1 \\ 0.0 & 0.1 & 0.1 & 0.3 & 0.5 \end{bmatrix}$.
- $\pi_{P1} = (0.180, 0.252, 0.184, 0.278, 0.106)$.
- $\pi_Q = (0.143, 0.190, 0.167, 0.357, 0.143)$.
- We can check that: $\pi_{P1} <_{st} \pi_Q$.
- Expectation: 1.87 for $P1$ and 2.16 for $v(P1)$.

Irreducibility of Q

- Due to the subtraction operations, some elements of $v(P)$ may be zero even if the corresponding elements in P are non zero.
- It may happen that matrix $v(P)$ computed by Vincent's algorithm is not irreducible, even if P is irreducible.
- If matrix $v(P)$ is reducible, it has one essential class of states. It is still possible to compute the steady-state distribution for this class.

- New algorithms which do not delete transitions while computing the bound (see IMSUB below and the Patterns).
- A necessary and sufficient condition on P to obtain an irreducible matrix.

IMSUB

Construction of an st-monotone upper bounding DTMC, Q without transition deletion, ϵ constant $0 < \epsilon < 1$

$Q_{1,n} = P_{1,n}$;
For $i = 2$ **to** n **Do** $Q_{i,n} = \max(P_{i,n}, Q_{i-1,n})$;
For $j = n-1$ **downto** 2 **Do**
  $Q_{1,j} = P_{1,j}$;
  For $i = 2$ **to** n **Do**
    $Q_{i,j} = \max(0, \max(\sum_{k=j}^{n} P_{i,k}, \sum_{k=j}^{n} Q_{i-1,k})) - \sum_{k=j+1}^{n} Q_{i,k}$;
    If $(Q_{i,j} = 0)$ **and** $(\sum_{k=j+1}^{n} Q_{i,k} < 1)$ **and** $((P_{i,j} > 0)$ **or** $(i = j-1))$
      then $Q_{i,j} = \epsilon \times (1 - \sum_{k=j+1}^{n} Q_{i,k})$;
  End
End
For $i = 1$ **to** n **Do** $Q_{i,1} = 1 - \sum_{k=2}^{n} Q_{i,k}$;

Theorem and Example

- **Theorem 2** *Let P be an irreducible finite stochastic matrix. Matrix Q computed from P with IMSUB is irreducible if and only if*
  - $P(1,1) > 0$,
  - *every row of the lower triangle of matrix P contains at least one positive element.*

$$P = \begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.2 & 0.0 \\ 0.1 & 0.7 & 0.1 & 0.0 & 0.1 \\ 0.2 & 0.1 & 0.5 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.7 & 0.3 \\ 0.0 & 0.2 & 0.2 & 0.1 & 0.5 \end{bmatrix} \quad Q = \begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.2 & 0.0 \\ 0.1 & 0.6 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.5 & 0.1 & 0.1 \\ \boxed{0.0} & \boxed{0.0} & \boxed{0.0} & 0.7 & 0.3 \\ 0.0 & 0.0 & 0.0 & 0.5 & 0.5 \end{bmatrix}$$

- States 0, 1 and 2 are transient.

Optimality

- **Theorem 3 (Optimality)** *Vincent's algorithm provides the smallest st-monotone upper bound for a matrix P: i.e. if we consider U another st-monotone upper bounding DTMC for P then $v(P) <_{st} U$ [1].*
- Proof based on properties of (max,+) equations.
- However bounds on the probability distributions may still be improved.
- The former theorem only states that Vincent's algorithm provides the smallest matrix according to the st-ordering of matrices.

Lower Bound

- Based on the same relations.
- Consider another ordering for the index of the rows and the columns.

$$\begin{array}{lcl} n & \rightarrow & 1 \\ n-1 & \rightarrow & 2 \\ & \ldots & \\ 1 & \rightarrow & n \end{array}$$

- Another operator (min instead of max).

Lower Bound

Construction of lower bound $Q <_{st} P$ and Q is $<_{st}$ monotone

Column 1:

$Q_{n,1} = P_{n,1}$;

For $i = n-1$ **downto** 1 **Do** $Q_{i,1} = \max(P_{i,1}, Q_{i+1,1})$;

Column j, $2 \leq j \leq n-1$:

For $j = 2$ **to** $n-1$ **Do**

  $Q_{n,j} = P_{n,j}$;

  For $i = n-1$ **downto** 1 **Do**

    $Q_{i,j} = \max(\sum_{k=1}^{j} P_{i,k}, \sum_{k=1}^{j} Q_{i+1,k}) - \sum_{k=1}^{j-1} Q_{i,k}$;

  End

End

Column n:

For $i = 1$ **to** n **Do** $Q_{i,n} = 1 - \sum_{k=1}^{n-1} Q_{i,k}$;

Lower Bound

Construction of lower bound $Q <_{st} P$ and Q is $<_{st}$ monotone

Column n:

$Q_{n,n} = P_{n,n}$;

For $i = n-1$ **downto** 1 **Do** $Q_{i,n} = \min(P_{i,n}, Q_{i+1,n})$;

Column j, $n-1 \leq j \leq 2$:

For $j = n-1$ **downto** 2 **Do**

$Q_{n,j} = P_{n,j}$;

For $i = n-1$ **downto** 1 **Do**

$Q_{i,j} = \min(\sum_{k=j}^{n} P_{i,k}, \sum_{k=j}^{n} Q_{i+1,k}) - \sum_{k=j+1}^{n} Q_{i,k}$;

End

End

Column 1:

For $i = 1$ **to** n **Do** $Q_{i,1} = 1 - \sum_{k=2}^{n} Q_{i,k}$;

Time and Space complexity

- $v(P)$ is, in general, as difficult as P to analyze.
- matrix $v(P)$ may have many more positive elements than matrix P and it may be even completely filled.

$$P4 = \begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.1 & 0.1 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \quad Q = \begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.1 & 0.1 \\ 0.5 & 0.2 & 0.1 & 0.1 & 0.1 \\ 0.5 & 0.2 & 0.1 & 0.1 & 0.1 \\ 0.5 & 0.2 & 0.1 & 0.1 & 0.1 \\ 0.5 & 0.2 & 0.1 & 0.1 & 0.1 \end{bmatrix}$$

Methodology for simplification

- Use the inequalities (degree of freedom) and build a matrix simpler to analyze.
- Easy to solve : matrices with structural or numerical properties (Pattern, Class C) or smaller matrices (lumpability, censored MC).
- Use ad-hoc algorithms for the numerical resolution of structured matrices or usual algorithms when the size of the bounding chain is small enough.
- No new assumptions on P.

Ordinary lumpability

- Used by Truffet with st-comparison to model ATM switch [48].
- Lumpability implies a state space reduction. (decomposition of the chain into macro-states)
- **Definition 7 (ordinary lumpability)** *Let X be an irreducible finite DTMC, Q its matrix, let A_k be a partition of the states. X is ordinary lumpable according to A_k, iff for all states e and f in the same arbitrary macro state A_i, we have:*

$$\sum_{j \in A_k} q_{e,j} = \sum_{j \in A_k} q_{f,j} \quad \forall \;\; macro-state \;\; A_k$$

- Ordinary lumpability constraints are consistent with st-monotonicity.
- An algorithm is proposed by Truffet [48].

Truffet's algorithm

- Assume that states are ordered according to the macro-state partition.
- Ordinary lumpability = constant row sum for the block
- The algorithm computes the matrix row by row with some particular work for block boundaries.
- Due to st-monotonicity, the maximal row sum is reached for the last row of the block (except for the last non-zero block).
- The values of the lumped matrix are obtained for the last row sum of a block.

Example

- $$P6 = \left[\begin{array}{cc|ccc} 0.5 & 0.2 & 0.2 & 0.0 & 0.1 \\ 0.2 & 0.4 & 0.2 & 0.2 & 0.0 \\ \hline 0.2 & 0.3 & 0.1 & 0.1 & 0.3 \\ 0.1 & 0.2 & 0.3 & 0.4 & 0 \\ 0.3 & 0.3 & 0.3 & 0 & 0.1 \end{array}\right].$$
- We divide the state-space into two macro-states: $(1, 2)$ and $(3, 4, 5)$.

- The bounding matrix and the row sums for the first block:

$$\left[\begin{array}{cc|ccc} 0.5 & 0.2 & 0.2 & 0.0 & 0.1 \\ 0.2 & 0.4 & 0.2 & 0.1 & 0.1 \\ \hline & & & & \\ & & & & \\ & & & & \end{array}\right] \quad \begin{array}{c} 0.3 \\ 0.4 \\ \hline \\ \\ \\ \end{array}$$

- The lumpable matrix and the lumped one:

$$\left[\begin{array}{cc|ccc} 0.4 & 0.2 & 0.3 & 0.0 & 0.1 \\ 0.2 & 0.4 & 0.2 & 0.1 & 0.1 \\ \hline & & & & \\ & & & & \end{array}\right] \quad \left[\begin{array}{cc} 0.6 & 0.4 \\ & \end{array}\right]$$

Various implementations

- LMSUB: Sparse matrix implementation of Truffet's algorithm [13].
- LIMSUB: add the irreducibility constraint (as IMSUB) [23].
- SAN2LMSUB: the input is a SAN (or a sum of tensor products). The output is a sparse matrix [26].

Censored Markov Chains

- Consider a DTMC with finite state space $S = E \cup E^c$, $E \cap E^c = \emptyset$.
- The censored DTMC with censoring set E watches the chain when it is in block E.
- For the steady-state, equivalent to the stochastic complement proposed by Meyer in [37].

  Consider a block decomposition of Q: $\begin{pmatrix} Q_E & Q_{EE^c} \\ Q_{E^cE} & Q_{E^c} \end{pmatrix}$.
- The stochastic complement matrix for block E:
  $S = Q_E + Q_{EE^c}(I - Q_{E^c})^{-1}Q_{E^cE}$.

- Q and Q_{E^c} is in general very large, so it is difficult to compute $(I - Q_{E^c})^{-1}$.
- $I - Q_{E^c}$ is not singular if Q is not reducible [37].
- Deriving bounds on S may be interesting.
- $\pi_S = \pi_S S$ with $\sum \pi_S = 1$

  π_S is the conditional steady-state probabilities for block E given that the DTMC is in block E

  $\pi_S = \pi_E / \sum \pi_E$.

Bounds for Censored Chains

- How to obtain stochastic bounds without computing $(I - Q_{E^c})^{-1}$?
- Avoid to build Q_{E^c} during the generation of the model.
- – Construct $\overline{S}$ such that $S <_{st} \overline{S}$.
  – Construct the monotone bound for $\overline{S}$ by Vincent's algorithm (R).
  – $\overline{S} <_{st} R$ and R is $<_{st}$-monotone.
  $\pi_S <_{st} \pi_R$

- The simplest way [47] is to put the slack probability
  to the **last** column for the **upper** bounding case,
  to the **first** column for the **lower** bounding case.
- Better repartition of the slack probability : DPY algorithm [22]

DPY Algorithm

Construction of an upper bounding stochastic matrix $\overline{S} : S <_{st} \overline{S}$;

Let $A_{1 \le i \le n_A, 1 \le j \le n_A}$ *denote* Q_E *and* $A_{n_A+1 \le i \le n, 1 \le j \le n_A}$ *denote* $Q_{E^c E}$

For $i = 1$ **to** n_A **Do** $\Delta_i = 1 - \sum_{j=1}^{n_A} A_{i,j}$;

last column: n_A*:*

For $l = n_A + 1$ **to** n **Do** $V_l = \frac{A_{l,n_A}}{\sum_{k=1}^{n_A} A_{l,n_A}}$;

$c = max_{n_A+1 \le l \le n} V_l$;

For $i = 1$ **to** n_A **Do**

$\overline{S}_{i,n_A} = A_{i,n_A} + \Delta_i c$; $\Delta_i = \Delta_i - \Delta_i c$; **End**

For $j = n_A - 1$ **downto** 1 (*column* j)

For $l = n_A + 1$ **to** n **Do** $V_l = \frac{\sum_{k=j}^{n_A} A_{l,k}}{\sum_{k=1}^{n_A} A_{l,k}}$;

$c = max_{n_A+1 \le l \le n} V_l$;

For $i = 1$ **to** n_A **Do**

$\overline{S}_{i,j} = A_{i,j} + \Delta_i c$; $\Delta_i = \Delta_i - \Delta_i c$; **End**

End

Example

$$Q = \left[\begin{array}{ccc|cc} & \mathbf{Q}_E & & \mathbf{Q}_{EE^c} & \\ 0.1 & 0.2 & 0.4 & 0.2 & 0.1 \\ 0.3 & 0.1 & 0 & 0.4 & 0.2 \\ 0.1 & 0 & 0 & 0.6 & 0.3 \\ \hline 0.1 & 0.2 & 0 & 0.3 & 0.4 \\ 0.2 & 0.4 & 0.2 & 0.1 & 0.1 \\ & \mathbf{Q}_{E^cE} & & \mathbf{Q}_{E^c} & \end{array}\right]$$

Consider block $\mathbf{Q}_E$

$$S = \begin{bmatrix} 0.1831 & 0.3661 & 0.4508 \\ 0.4661 & 0.4322 & 0.1017 \\ 0.3492 & 0.4983 & 0.1525 \end{bmatrix} \quad \overline{S} = \begin{bmatrix} 0.175 & 0.350 & 0.475 \\ 0.450 & 0.400 & 0.150 \\ 0.325 & 0.450 & 0.225 \end{bmatrix}$$

Example (cont.)

- $S \leq_{st} \overline{S}$
- Monotone and upper-bounding matrix of $\overline{S}$:

$$R = \begin{bmatrix} 0.1750 & 0.3500 & 0.4750 \\ 0.1750 & 0.3500 & 0.4750 \\ 0.1750 & 0.3500 & 0.4750 \end{bmatrix}$$

- $\pi_S = [0.3420, 0.4250, 0.2330]$
  $\pi_R = [0.1750, 0.3500, 0.4750]$

$$\pi_S <_{st} \pi_R$$

Example (cont.)

Truffet's algorithm gives $\overline{S'}$ and we obtain R' by Vincent's algorithm:

$$\overline{S'} = \begin{bmatrix} 0.1000 & 0.2000 & 0.7000 \\ 0.3000 & 0.1000 & 0.6000 \\ 0.1000 & 0.0000 & 0.9000 \end{bmatrix} \quad R' = \begin{bmatrix} 0.1000 & 0.2000 & 0.7000 \\ 0.1000 & 0.2000 & 0.7000 \\ 0.1000 & 0.0000 & 0.9000 \end{bmatrix}$$

$$S <_{st} \overline{S} <_{st} \overline{S'}$$

$$\pi_{R'} = [0.1000, 0.0250, 0.8750]$$

$$\pi_S <_{st} \pi_R <_{st} \pi_{R'}$$

Pattern

- A matrix notation to define the graph of the bounding Markov chain.
- A pattern is a matrix of symbols. This matrix has the same size as the original matrix (simplification of the structure, we do not modify the number of states).
- 5 symbols :
  - 0: the arc must not exist,
  - 1: the transition must have a positive probability,
  - w, s: if the transition exists in P, it must exist in the bounding matrix (error handling differs),
  - *: no constraint.

Bušić's algorithm for Pattern

- Input: a pattern and a stochastic matrix P.
- Output: an upper bound monotone matrix consistent with the pattern or an error message (it is not possible to obtain such a bound).
- Complexity: at worst quadratic.
- a row by row algorithm.
- If you design a new numerical technique for a family of Markov chain characterized by their graphs, you design the pattern and you obtain (for free) a bounding algorithm already proved.
- Already known patterns: Upper-Hessenberg, Single Input Macro State MC, Stochastic Complement with block Q_{E^c} triangular, Vincent's algorithm and IMSUB.

Example: Upper-Hessenberg Pattern

- **Definition 8** *A matrix H is said to be upper-Hessenberg if and only if $H_{i,j} = 0$ for $i > j + 1$.*
- The resolution by recursion for these matrices requires $o(m)$ operations (Stewart [43]).
- Upper-Hessenberg property is consistent with comparison and monotonicity.
- Pattern : $\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$.

Bušić's algorithm for Pattern

Construction of a st-monotone upper bounding DTMC, Q consistent with pattern T; ϵ constant $0 < \epsilon < 1$

For $i = 1$ **to** n **Do**
  last = -1;
  For $j = n$ **to** 1 **Do**
    $sum = \sum_{k=j}^{n} P_{i,k}$;
    If $(i > 1)$ **then** $sum = \max(sum, \sum_{j=k}^{n} Q_{i-1,k})$;
    If $(j < n)$ **then** $Q_{i,j} = \max(0, sum - \sum_{j=k+1}^{n} Q_{i,k})$;
    else $Q_{i,j} = sum$;
    Switch $T_{i,j}$ **Do**
      See Next Slides
    End
  End
End

Bušić's algorithm-cont.

Case-block for symbols 0 and 1.

Case 0

If $Q_{i,j} > 0$ **then**

If $last > 0$ **then**

$Q_{i,last} = Q_{i,last} + Q_{i,j};$

$Q_{i,j} = 0;$

else STOP : NOT CONSISTENT !;

End

Case 1

$last = j;$

If $Q_{i,j} = 0$ **then**

If $\sum_{k=j+1}^{n} Q_{1,k} < 1$ **then** $Q_{i,j} = \epsilon \times (1 - \sum_{k=j+1}^{n} Q_{i,k});$

else STOP : NOT CONSISTENT !;

End

Bušić's algorithm-cont.

Case-block for symbols $*$, w *and* s.

Case $*$

$last = j;$

Case s, w

$last = j;$

If $P_{i,j} > 0$ **then**

If $\sum_{k=j+1}^{n} Q_{1,k} < 1$ **then** $Q_{i,j} = \epsilon \times (1 - \sum_{k=j+1}^{n} Q_{i,k});$

elsif $T_{i,j} = s$ **then** STOP : NOT CONSISTENT !;

End

Getting more from the classic framework

- Improving accuracy.
- Continuous Time Markov Chain.
- Transient analysis of rewards.
- Absorbing DTMC.
- Qualitative properties.
- Worst Case Analysis.

Improving accuracy

- Apply some transformations [19] on P before Vincent's algorithm.
- First, $\alpha(P, \delta) = (1-\delta)Id + \delta P$, for $\delta \in (0, 1)$.
- It has no effect on the steady-state distribution.
- It has a large influence on the effect of Vincent's algorithm.
- **Theorem 4** *Let P be a DTMC, and two different values $\delta_1, \delta_2 \in (0, 1)$ such that $\delta 1 < \delta 2$, Then $\pi_{v(\alpha(P,\delta 1))} <_{st} \pi_{v(\alpha(P,\delta 2))} <_{st} \pi_{v(P)}$.*

A good value for δ

- **Definition 9** *A stochastic matrix is said to be row diagonally dominant (RDD) if all of its diagonal elements are greater than or equal to 0.5.*
- **Corollary 1** *Let P be a RDD DTMC, then $v(P)$ and $v(\alpha(P))$ have the same steady-state probability distribution.*
- Idea : For a RDD matrix, the diagonal serves as a barrier for the perturbation moving from the upper-triangular part to the strictly lower-triangular part $v(P)$.
- $\delta = 1/2$ is sufficient to make an arbitrary stochastic matrix RDD.
- Thus the transformation $P/2 + Id/2$ provides the best bound for these linear transformations.

Polynomials

- To obtain more accurate bounds.
- **Definition 10** *Let $\mathcal{D}$ be the set of polynomials $\Phi()$ such that $\Phi(1) = 1$, Φ different of Identity, and all the coefficients of Φ are non negative.*
- **Proposition 1** *Let $\Phi()$ be an arbitrary polynomial in $\mathcal{D}$, then $\Phi(P)$ has the same steady-state distribution than P.*
- **Theorem 5** *Let Φ be an arbitrary polynomial in $\mathcal{D}$, Algorithm 1 applied on $\Phi(P)$ provides a more accurate bound than the steady-state distribution of $v(P)$ i.e.:*

$$\pi_P <_{st} \pi_{v(\Phi(P))} <_{st} \pi_{v(P)}.$$

- For a stochastic interpretation of this result and a proof based on linear algebra see [20].

Example

- Polynomials with larger degree may give more accurate bounds. This is illustrated in the example below.

$$P3 = \begin{bmatrix} 0.1 & 0.2 & 0.4 & 0.3 \\ 0.2 & 0.3 & 0.2 & 0.3 \\ 0.1 & 0.5 & 0.4 & 0 \\ 0.2 & 0.1 & 0.3 & 0.4 \end{bmatrix}$$

- We study the polynomials $\phi(X) = X/2 + 1/2$ and $\psi(X) = X^2/2 + 1/2$.

$$\phi(P3) = \begin{bmatrix} 0.55 & 0.1 & 0.2 & 0.15 \\ 0.1 & 0.65 & 0.1 & 0.15 \\ 0.05 & 0.25 & 0.7 & 0 \\ 0.1 & 0.05 & 0.15 & 0.7 \end{bmatrix}$$

$$\psi(P3) = \begin{bmatrix} 0.575 & 0.155 & 0.165 & 0.105 \\ 0.08 & 0.63 & 0.155 & 0.135 \\ 0.075 & 0.185 & 0.65 & 0.09 \\ 0.075 & 0.13 & 0.17 & 0.625 \end{bmatrix}$$

- Then, we apply operator v to obtain the bounds:

$$v(\phi(P3)) = \begin{bmatrix} 0.55 & 0.1 & 0.2 & 0.15 \\ 0.1 & 0.55 & 0.2 & 0.15 \\ 0.05 & 0.25 & 0.55 & 0.15 \\ 0.05 & 0.1 & 0.15 & 0.7 \end{bmatrix}$$

$$v(\psi(P3)) = \begin{bmatrix} 0.575 & 0.155 & 0.165 & 0.105 \\ 0.08 & 0.63 & 0.155 & 0.135 \\ 0.075 & 0.185 & 0.605 & 0.135 \\ 0.075 & 0.13 & 0.17 & 0.625 \end{bmatrix}$$

- And,

$$v(P3) = \begin{bmatrix} 0.1 & 0.2 & 0.4 & 0.3 \\ 0.1 & 0.2 & 0.4 & 0.3 \\ 0.1 & 0.2 & 0.4 & 0.3 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{bmatrix}$$

- Finally, we compute the steady-state distributions:

$$\begin{cases} \pi_{v(P3)} = (0.1, 0.2, 0, 3667, 0.3333) \\ \pi_{v(\phi(P3))} = (0.1259, 0.2587, 0, 2821, 0.3333) \\ \pi_{v(\psi(P3))} = (0.1530, 0.2997, 0, 2916, 0.2557) \\ \pi_{P3} = (0.1530, 0.3025, 0, 3167, 0.2278) \end{cases}$$

- Clearly, bounds obtained by ψ are more accurate than the other bounds.

But. . .

- But it is not always true that the higher the degree the more accurate the bounds. . .
- See for instance antimonotone DTMC (see [20]).
- Finding good polynomials for preprocessing is still an open problem.

Continuous Time Markov Chain

- The theory of comparison and monotonicity of CTMC exists.
- Less constraints on the matrices.
- A birth and death process is monotone while a tridiagonal DTMC may be monotone or not depending of the values.
- An algorithm exists (TVP [46] but the input is a Stochastic Automata Network without synchronizations).

Basic constraints for an algorithm on CTMC

- **Theorem 6 (Comparison)** *Let X and Y two CTMC with transition rate matrix A and B. If the following two conditions are satisfied:*
  1. $X_0 <_{st} Y_0$
  2. *For all i, j and m such that $i \leq j$ and $m \leq i$ or $m > j$ we have:*

  $$\sum_{k>m} A_{i,k} \leq \sum_{k \geq m} B_{j,k}$$

  then $X_t <_{st} Y_t$, $t > 0$.
- Slightly more difficult to make proofs (because we have exceptions on the diagonal elements).

Uniformization for steady-state analysis

- Let Q be the transition rate matrix and let $\gamma = max(-Q(i,i))$ and $\epsilon \geq 0$.
- Uniformization $u_\epsilon(Q) = \frac{Q}{\gamma+\epsilon} + Id$.
- $u_\epsilon(Q)$ and Q have the same steady-state.
- A natural question: Is it more accurate to compute the bound on the CTMC or on the uniformized version of the CTMC using Vincent's algorithm ?

CTMC or DTMC

- If the uniformized version of the CTMC is RDD, the bounds are the same, otherwise the bounds for the CTMC are more accurate (or equal).
- Uniformization to RDD : it is sufficient to have $\epsilon = \gamma$.
- Use the RDD-Uniformization and the algorithms on DTMC.

Analysis of Rewards at time t

- For CTMC use the uniformization formula, π_0 is the initial distribution, λ the uniformization factor:

$$P(X_t \in U) = e^{-\lambda t} \sum_{n=0}^{\infty} \frac{(\lambda t)^n}{n!} \pi_0 P_\lambda^n 1_U$$

- As usual, we truncate the summation index to N_β to obtain a proved accuracy smaller than β.
- If $P <_{st} Q$ and Q is monotone, then $P^n <_{st} Q^n$.
- Check that 1_U is increasing.
- For DTMC, matrix-product operation.

Algorithms for DTMC

- Ordinary Lumpability = Strong Aggregation.
- The lumped process is still Markov.
- Truffet's first algorithm, LMSUB and LIMSUB provide upper bounds for transient distributions
- Transient distributions of class C matrices also have closed form [6].
- Pattern based and censored based bounds are still under study for analysis of transient distributions and rewards.

Analysis of absorbing time

- **Theorem 7** *[3] Let X and Y two DTMC on state space $0..n$ absorbing in n (only one absorbing state), with stochastic matrices P and Q assume that:*
  1. $X_0 = Y_0$
  2. *P or Q is st-monotone*
  3. $P <_{st} Q$

  then $T_Y <_{st} T_X$ where T_X is the absorbing time in n for chain X.
- The output of LMSUB may be a lumped matrix which is still absorbing (some technical conditions to check).
- It is much easier to compute the fundamental matrix on the lumped chain.

Qualitative Properties

- A recent application (Valuetools 2007 [11]).
- How to prove that an absorbing time (or a st-st reward) is increasing with a parameter of the model ?
- A simple example rather than a general theory.
- How to prove some algorithms based on Markov chains and mean interaction.

End to end delay with SP deflection routing

- Deflection routing: used when it is impossible to store packets waiting for the best output (typically all optical switch).
- Shortest Path Deflection routing: try shortest paths but use deflection when the number of packets exceeds the link capacity.
- Major Assumption: Topology + Independence of packets + Uniform distribution for the O-D imply an aggregated Markov chain whose state is the distance to the destination.
- 0 is an absorbing state.

Effect of a deflection

- **Definition 11 (Symmetric Graph)** *A graph $G = (V, E)$ is symmetric iff for all i and j nodes in V, if (i, j) is a directed edge in E, (j, i) is also in E.*
- **Property 2** *In a symmetric graph, the deflected packet originally at distance k can jump at distance $k - 1$ or $k + 1$ or is still at distance k (because of the shortest-path deflection routing).*
- Let p (unknown) be the deflection probability and $R(p)$ the transition matrix.

Topology

- An odd ring
- In the example, the size of the graph (sz) is 7.
- Thus the states of the chain are $0, 1, 2, 3$.

Transitions for an odd ring

- If $k = 0$ stay in the same state.
- If the packet is not deflected: transition form k to $k-1$ with probability $1-p$.
- If the packet is deflected: transition from k to $k+1$ except when $k = sz$ where the packet is kept at distance sz after deflection (due to the odd ring topology).
- $$R(p) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1-p & 0 & p & 0 \\ 0 & 1-p & 0 & p \\ 0 & 0 & 1-p & p \end{bmatrix}.$$

Initial Distribution

- Uniform destination and source (but source $\neq$ destination).
- Two nodes at each distance.
- Initial distribution for the ring with 7 nodes: $(0, 1/3, 1/3, 1/3)$.

Properties

- The matrix is monotone for all value of p; this is always true for an odd ring and always false for an even ring.
- If $p1 > p2$ then $R(p2) <_{st} R(p1)$.
- Absorption time in 0: end to end delay in the network (without taking into account the insertion delay at the interface).
- $E(X(p)) < \infty$ if $p < 1$.

Main Results

- If $p1 > p2$ $X(p1) <_{st} X(p2)$.
- $E(X(p))$ is increasing with p.
- If we are able to find bounds on p, we can derive bounds on $X(p)$.
- For instance $pmin \leq p \leq pmax$ implies than $E(X(pmin)) \leq E(X(p)) \leq E(X(pmax))$.

A proved approximate analysis

- Little's law: $E(N) = \lambda E(X(p))$.
- where λ is the accepted arrival rate.
- Link Utilization: $u = \frac{E(N)}{2sz}$ because a directed ring with sz nodes has $2sz$ directed edges.
- This gives an increasing function f such that $u = f(p)$.

Assumptions

- You know another model which provides $p = g(u)$ such that:
- g is increasing.
- and $g(1) < 1$. Indeed a conflict between k packets give $k - 1$ deflection.
- Thus you have a fixed point system $u = f(p)$ and $p = g(u)$.

Proving the existence of a solution

- f and g are increasing.
- $g(1) < 1$.
- f and g are upper-bounded.
- **Theorem 8** *As the sequence* $(p_0 = 0, p_{i+1} = g(f(p_i)))$ *is increasing and upper-bounded, it has a limit which is a solution of the fixed point system.*

First proved algorithm

- Iterative Algorithm: follow the sequence defined in the theorem.
- The convergence is proved.
- Stopping criteria: $|p_{i+1} - p_i| \leq \epsilon$ does not mean that $|p^* - p_i| \leq \epsilon$ (p^* is the limit).

A better algorithm

- Based on a dichotomic search.
  1. Begin with interval $a = 0$ and $b = g(1)$.
  2. Let c=(a+b)/2 and compute $f(g(c))$.
  3. If $|b - a| < \epsilon$ Stop.
  4. If $f(g(c)) > c$ let $a = c$ and go to step 2.
  5. else let $b = c$ and go to step 2.
- Also proved and the solution is in the interval $[a, b]$.

Two qualitative results

- Some performance indices are increasing functions of the parameters.
- Proof of the convergence of a method based on the iterative solution of subproblems if one of the subproblems is the analysis of a Markov chain.
- Is it possible to prove some well known approximate iterative methods ?

Worst Case Analysis

- For analysis of stochastic matrices which are not completely specified.
- For instance, the transition probabilities are not exactly known; we just give some intervals.
- $$M = \begin{bmatrix} 0 & 1-a-b & b & a \\ 1-a/2 & a/2 & 0 & a/2 \\ 1-b/2 & 0 & b/2 & b/2 \\ 1-a-b & 0 & 0 & a+b \end{bmatrix}$$
  with $1/3 \leq a \leq 1/2$ and $1/4 \leq b \leq 1/3$.
- For steady-state analysis see recent paper by Buchholz [8] based on polyhedral theory.

A stochastic approach

- Allows more general results.
- Transient and steady state analysis.
- Time to Failure (absorption).
- Based on stochastic ordering and monotonicity.
- We only consider here matrices where elements are in intervals (a different approach is used in the section on icx-ordering).

Partially defined DTMCs

- Consider a set of stochastic matrices $P \in \mathcal{P}(L, U)$.
- $L \leq_{el} P \leq_{el} U, \quad \forall P \in \mathcal{P}$.
- Construction of extreme stochastic matrices $\overline{P}$ and $\underline{P}$ by Truffet [47] such that $\underline{P} <_{st} P <_{st} \overline{P}, \quad \forall P \in \mathcal{P}$

Truffet's 2nd Algorithm

Construction of the extreme upper bound $\overline{P}$ for the set $\mathcal{P}(L, U)$

For $i = 1$ **to** n **Do**

$\Delta_i = 1 - \sum_{j=1}^{n} L_{i,j}$;

For $j = n$ **downto** 1 **Do**

$\delta = \min(\Delta_i, (U_{i,j} - L_{i,j}))$;

$\overline{P}_{i,j} = L_{i,j} + \delta; \qquad \Delta_i = \Delta_i - \delta$;

End

End

- Lower Bound obtained by adding Δ from beginning by the first column
- If $U_{i,*} = L_{i,*} + \Delta_i \ \forall i$, it leads to complete in the last column for the upper bound and in the first column for the lower bound
- A similar algorithm presented by Haddad and Moreaux for substochastic matrices to improve the polyhedral approach [29].

Optimality

- Let $\overline{Q}$ and $\underline{Q}$ be monotone matrices obtained by Vincent's algorithm for input matrices $\overline{P}$ and $\underline{P}$.
- $\overline{Q}$ and $\underline{Q}$ are optimal monotone bounds for the set $\mathcal{P}(L, U)$:
  If monotone stochastic matrices A, B exist such that
  $A <_{st} P <_{st} B \quad \forall P \in \mathcal{P}(L, U)$
  then $A <_{st} \underline{Q}$ and $\overline{Q} <_{st} B$
- Stochastic bounds on the transient and steady-state distributions for the set of matrices defined by $\mathcal{P}(L, U)$:
  $\Pi_{\underline{Q}}(t) <_{st} \Pi_P(t) <_{st} \Pi_{\overline{Q}}(t) \quad \forall t,\ \forall P \in \mathcal{P}(L, U)$

Beyond the classic method

- Partial ordering on the state space
  - Monotony for free
  - Non monotone systems
- Increasing convex ordering (icx).

Increasing Convex Ordering

- A variability ordering.
- More complex than the usual st ordering.
- More accurate than st ordering when one deals with random variables.
- If $X <_{st} Y$ and $E(X) = E(Y)$ then X and Y are identically distributed.
- It is possible to consider the set of random variables with the same expectation and find the maximal or minimal r.v. according to the icx ordering.

Increasing Convex Ordering

- **Definition 12** *Let X and Y be two random variables taking values on a totally ordered space space. Then we say that X is smaller than Y in the increasing convex sense (icx),*

$$X <_{icx} Y \quad if \quad E(f(X)) \leq E(f(Y))$$

for all increasing and convex functions f whenever the expectations exist.

- Thus "st" ordering (defined by increasing functions) implies "icx" ordering (defined by increasing and convex).

On discrete state space

$$X <_{icx} Y \iff \sum_{k=i}^{n}(k-i+1)\ x_k \leq \sum_{k=i}^{n}(k-i+1)\ y_k, \quad \forall i$$

$$\iff \left\{ \begin{array}{rcr} x_n & \leq & y_n \\ x_{n-1}+2x_n & \leq & y_{n-1}+2y_n \\ x_{n-2}+2x_{n-1}+3x_n & \leq & y_{n-2}+2y_{n-1}+3y_n \\ & \ldots & \\ x_1+2x_2+\ldots+nx_n & \leq & y_1+2y_2+\ldots+ny_n \end{array} \right.$$

Example

- Three probability vectors: $x = (0.5, 0.1, 0.1, 0.3)$, $y = (0.3, 0.2, 0.2, 0.3)$, and $z = (0.3, 0.2, 0.4, 0.1)$
- $x <_{icx} y$ as
  - $0.3 \leq 0.3$ and $0.1 + 2 * 0.3 \leq 0.2 + 2 * 0.3$
  - $0.1 + 2 * 0.1 + 3 * 0.3 \leq 0.2 + 2 * 0.2 + 3 * 0.3$
- The vectors x and z are not icx-comparable as
  - $x_3 = 0.3 > 0.1 = z_3$, but
  - $x_1 + 2x_2 + 3x_3 = 1.2 < 1.3 = z_1 + 2z_2 + 3z_3$.

icx-monotone DTMC

- Much harder constraints.
- Ben Mamoun's characterization for finite DTMC:

  P is icx-monotone iff $Z_{icx} P K_{icx} \geq 0$ component-wise with:

$$Z_{icx} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 1 & -2 & 1 \end{bmatrix} \quad K_{icx} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 2 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & n-1 & n-2 & \dots & 1 \end{bmatrix}$$

No Optimal Bound for icx ordering of DTMC

- Consider $P = \begin{bmatrix} 0.5 & 0.4 & 0.1 \\ 0.3 & 0.3 & 0.4 \\ 0.1 & 0.4 & 0.5 \end{bmatrix}$,
- and $U1$ and $U2$ which are icx monotone upper bound of P:

$$U1 = \begin{bmatrix} 0.5 & 0.4 & 0.1 \\ 0.3 & 0.3 & 0.4 \\ 0.1 & 0.2 & 0.7 \end{bmatrix} \quad U2 = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.3 & 0.4 \\ 0.1 & 0.4 & 0.5 \end{bmatrix}$$

- It is not possible to prove an optimal bound Q such that $P <_{icx} Q$, $Q <_{icx} U1$ and $Q <_{icx} U2$.
- Indeed the last column of Q must be $(0.1, 0.4, 0.5)^t$ which is not convex.

Bušić's method for an icx monotone upper bound

- By column, from column n to 1.
- The computation of the upper bound matrix is based on the resolution of the following problem (for each column).
- Problem CV: Let a and b two vectors such that $0 \leq_{el} a \leq_{el} b$. Find a vector x increasing and convex such that $a \leq_{el} x \leq_{el} b$.
- Need two sequences ϕ^{st} and ϕ^{icx} built during the algorithm to provide successive values for a and b.

Details of Bušić's method

1. Solve Problem CV with $a = P_{*,n}$ and $b = (1, \ldots, 1)^t$. And $Q_{*,n} = x$, $\phi^{icx}_{*,n} = x$, $\phi^{st}_{*,n} = x$
2. For all column index from $n-1$ to 2 Solve Problem CV with:

   $$a = max(\phi^{icx}_{i,j}(P), \phi^{icx}_{i,j+1}(Q) + \phi^{st}_{i,j+1}(Q))$$

   and

   $$b = \phi^{icx}_{i,j+1}(Q) + 1$$

   And $\phi^{icx}_{i,j}(Q) = x$, $\phi^{st}_{i,j}(Q) = \phi^{icx}_{i,j}(Q) - \phi^{icx}_{i,j+1}(Q)$.
   Finally $Q_{i,j} = \phi^{st}_{i,j} - \phi^{st}_{i,j+1}$.
3. Row 1: Normalization.

Solving the vector problem

- Many heuristics (see Bušić's PHD [9]).
- None of them are optimal.
- Take care of the complexity.
- One must solve Problem CV n times.
- Avoid to obtain a trivial solution with the last column equal to $(1, \ldots, 1)^t$.

icx ordering for DTMC

- Proof that there is no optimal bound.
- Difficult to apply to a general matrix.
- Very accurate when the model is almost icx-monotone.

A Batch/D/1/N queue

- Buffer size for optical packet switch with constant packet size
- Without electronic conversion (no electronic buffer) : use Fiber Delay Loops instead
- Without wavelength conversion: 1 server per wavelength.
- K input links.
- ROM and ROMEO architectures (Alcatel)
- Batch/D/1/N queue
- We know the average arrival rate (easy to measure) and the maximal batch size K.
- Can we dimension the buffer ?

Steps of the analysis

- Note that the model is almost-icx monotone.
- Use icx-ordering.
- Find the worst arrival process according to icx-ordering and derive the Markov chain of the queue.
- Scale the chain to allow icx-comparison.
- Make the scaled Markov chain icx monotone.

Worst Case Arrival

- $A = (a_0, \ldots, a_K)$ = distribution of batch arrivals.
- $\alpha = E(A)$ is known.
- We assume: $N > K$ (engineering) and $\alpha < 1$ (stability).
- $\mathcal{F}_\alpha$ = the family of all distributions on the space $\{0, \cdots, N\}$ having expectation α
- icx-worst case distribution: $q = (\frac{N-\alpha}{N}, 0, \ldots, 0, \frac{\alpha}{N})$:
- **Property 3 (Maximal R.V. (see Shantikumar))**

$$q \in \mathcal{F}_\alpha \quad and \quad p \preceq_{icx} q, \; \forall p \in \mathcal{F}_\alpha$$

Matrix of the Chain

-

$$\boldsymbol{P} = \begin{pmatrix} a_0 & a_1 & \cdots & a_K & 0 & \cdots & 0 \\ a_0 & a_1 & \cdots & a_K & 0 & \cdots & 0 \\ 0 & a_0 & a_1 & \cdots & a_K & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & a_0 & a_1 & \cdots & a_K \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & a_0 & \sum_{i=1}^{K} a_i \end{pmatrix}$$

- A bound of the arrival rate is not sufficient.
- The matrix must be monotone (and $\boldsymbol{P}$ is not...).

4 Steps

1. Build an upper icx-bound $\boldsymbol{Q}$ for each row using the worst arrival process.

$$\boldsymbol{Q} = \begin{cases} Q_{0,0} = 1 - \frac{\alpha}{K} & Q_{0,K} = \frac{\alpha}{K} \\ 0 < i \leq N-K+1 & Q_{i,i-1} = (1-\frac{\alpha}{K}) & Q_{i,i+K-1} = \frac{\alpha}{K} \\ N-K+2 \leq i < N & Q_{i,i-1} = (1-\frac{\alpha}{N-i+1}) & Q_{i,N} = \frac{\alpha}{N-i+1} \\ Q_{N,N-1} = (1-\alpha) & Q_{N,N} = \alpha \end{cases}$$

   $\boldsymbol{Q}$ is not monotone

2. Modify matrix $\boldsymbol{Q}$: $t_\delta(\boldsymbol{Q}) = \delta\boldsymbol{Q} + (1-\delta)\boldsymbol{Id}$

   t_δ: same steady-state distribution, move some probability mass to the diagonal elements to allow step 4.

3. Apply the forward algorithm to make the last row of $t_\delta(\boldsymbol{Q})$ increasing and convex

4. Change diagonal and sub-diagonal elements to make final matrix $\boldsymbol{B}$ icx-monotone (only some of them)

$$\boldsymbol{B} = \begin{cases} B_{0,0} = 1 - \delta\frac{\alpha}{K} & B_{0,K} = \delta\frac{\alpha}{K} \\ 1 \leq i \leq N-K+1: \\ B_{i,i-1} = \delta(1-\frac{\alpha}{K}) & B_{i,i} = 1-\delta & B_{i,i+K-1} = \delta\frac{\alpha}{K} \\ N-K+2 \leq i < N: \\ B_{i,i-1} = f_i & B_{i,i} = e_i & B_{i,N} = \delta\frac{\alpha}{K}(i-N+K) \\ B_{N,N-1} = \delta(1-\alpha) & & B_{N,N} = 1-\delta+\delta\alpha \end{cases}$$

   where $e_i = 1 - \delta + \delta\alpha - (N-i+1)B_{i,N}$ and $f_i = 1 - e_i - B_{i,N}$.

Main result

Theorem 9 *Suppose that*

$$\delta \leq \frac{1}{1+\alpha U}, \tag{2}$$

where $U = \max_{r=2\ldots K-1} \frac{r(K-r+1)}{K}$. *Then,*

1. $\boldsymbol{B}$ *is a stochastic matrix.*
2. $\boldsymbol{B}$ *is irreducible.*
3. $\boldsymbol{Q} <_{icx} \boldsymbol{B}$.
4. $\boldsymbol{B}$ *is icx-monotone.*

Accuracy

- The perturbation added by the monotonicity constraint is relatively small (i.e. difference between st-st distribution of $\boldsymbol{Q}$ and $\boldsymbol{B}$).
- The main error comes from the main assumption (we ONLY know the average and the max batch size).
- What type of information can we add ? (p0: probability of an empty batch).

A numerical example

- A state dependent batch.
- Back-pressure mechanism. When the queue size is large, a signal is sent to the sources of traffic to avoid congestion and shape the traffic.
- Shaping: same average (not that important, we can reduce) and smaller variability.
- Smaller variability: smaller K.
- Threshold: 80% of the buffer size.

Average number of packets in the queue

	$\boldsymbol{S}$	$\boldsymbol{B}$	rel. error	$\boldsymbol{S}$	$\boldsymbol{B}$	rel. error
0.5	5.000e+00	5.000e+00	$< 10^{-15}$	5.00e+01	5.00e+01	2.7e-05
0.8	1.880e+01	1.880e+01	$< 10^{-15}$	1.93e+02	1.97e+02	1.5e-02
0.9	4.140e+01	4.140e+01	8.9e-09	3.69e+02	3.92e+02	6.3e-02
0.95	8.644e+01	8.645e+01	9.1e-05	5.45e+02	6.06e+02	1.1e-01
0.99	3.780e+02	3.984e+02	5.3e-02	7.95e+02	9.00e+02	1.3e-01

Table 1: Comparison of the mean queue length at the steady-state between the state dependent ($\boldsymbol{S}$) and the monotone upper bound ($\boldsymbol{B}$) for $N = 1000$, $K = 10$ and $K = 100$.

Is the model monotone ?

- From several years of practice: multidimensional models with a total ordering are not st-monotone.
- But with the natural partial ordering, multidimensional models are often monotone (with an intuitive definition of monotonicity) [10, 15].
- For instance Queuing networks (see Glasserman and Yao [27]).
- Also true for some families of Petri nets.
- If the model is monotone we do not need to build a monotone bound. We just have to compute an upper bound simpler to solve (a simpler task).

St-Ordering of DTMC on partially ordered space

- **Definition 13 (Massey)** *$X \preceq_{st} Y$ if and only if $P(X \in U) \leq P(Y \in U)$, for all increasing sets $U \subset S$.*
- **Definition 14 (Increasing Set)** *A subset $U \in S$ is called an increasing set if its indicator function $\mathbf{1}_U$ is increasing. Or if and only if $x \in U$ and $x \preceq_S y$ imply $y \in U$.*
- Finite totally ordered set $(S, \preceq_S)$, $|S| = n$, there are exactly n different increasing sets $U \neq \emptyset$.
- For partial order, we may have an exponential number of increasing sets: comparison of r.v. is not that simple.

Example

- Let $S = \{1, 2, 3, 4, 5\}$ and let the ordering relation $\preceq_A$ be defined as $1 \preceq_A i \preceq_A 5$, $\forall i \in S$. There are 9 increasing sets $U \neq \emptyset$: $\{5\}$, $\{2, 5\}$, $\{3, 5\}$, $\{4, 5\}$, $\{2, 3, 5\}$, $\{2, 4, 5\}$, $\{3, 4, 5\}$, $\{2, 3, 4, 5\}$ and S.

Example-cont.

- If we consider random variables X, Y and Z with distribution vectors:

$$x = (0.3, 0.3, 0.1, 0.1, 0.2),$$
$$y = (0.3, 0.1, 0.2, 0.1, 0.3),$$
$$z = (0.1, 0.2, 0.2, 0.1, 0.4),$$

- then, with ordering $\preceq_A$ on the states, we have $X \preceq_{st,A} Z$ and $Y \preceq_{st,A} Z$, but X and Y are not comparable in the $\preceq_{st,A}$-sense since $P(X = 5) = 0.2 < P(Y = 5) = 0.3$ but $P(X \in \{2, 5\}) = 0.5 > P(Y \in \{2, 5\}) = 0.4$.
- However, if we consider the total order $\preceq_B$ on S, $1 \preceq_B 2 \preceq_B \ldots \preceq_B 5$,
- We have $X \preceq_{st,B} Y \preceq_{st,B} Z$.

Monotonicity and Comparison

- **Proposition 2** *Let $\{X_t\}$ be a homogeneous DTMC on a partially ordered state space $(S, \preceq_S)$. The transition matrix P of $\{X_t\}$ is $\preceq_{st}$-monotone if for all $i, j \in S$,*

$$i \preceq_S j \implies P_{i,*} \preceq_{st} P_{j,*},$$

i.e. if $\sum_{k \in U} P_{i,k} \leq \sum_{k \in U} P_{j,k}$ for all increasing sets U.

- **Definition 15** *For transition matrices P and Q we say that $P \preceq_{st} Q$ if*

$$P_{i,*} \preceq_{st} Q_{i,*} \text{ for all } i \in S,$$

i.e. if $\sum_{k \in U} P_{i,k} \leq \sum_{k \in U} Q_{i,k}$ for all increasing sets U.

- The comparison will be harder but we hope to avoid the monotonicity

Example

-

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

- Partial order $1 \preceq_A 2 \preceq_A 5$ and $3 \preceq_A 4 \preceq_A 5$
- The chain is $\preceq_{st,A}$-monotone.
- $\pi_P = (0.182, 0.303, 0.115, 0.212, 0.188)$

Example-cont.

- Total Order $1 \preceq_B 2 \preceq_B 3 \preceq_B 4 \preceq_B 5$
- The chain is not monotone
- Best monotone upper bounding matrix computed by Vincent's algorithm:

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

- $\pi_Q = (0.125, 0.287, 0.115, 0.245, 0.228)$.

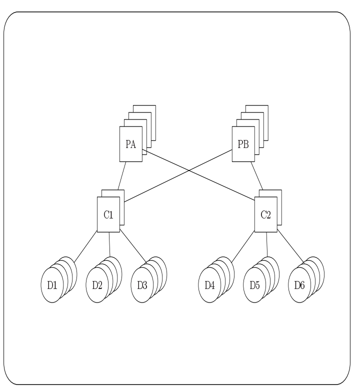
Is it really simpler ?

- If the chain is monotone, find some upper bound DTMC easier to solve (Algorithm LL: lumpable and larger).
- If the system is not monotone, we still have to find a monotone upper bound.
- The complexity is related to the number of increasing sets.
- Comparison of Stochastic matrix may be hard.
- And monotonicity may also be difficult.
- Still under study to find some simple families related to high level formalisms (Petri nets, Queueing Networks).
- New Idea: Modify the ordering of the states and the rewards ?

LL Algorithm

- To obtain a lumpable upper bound of a monotone DTMC.
- Designed to avoid the state-space and the matrix generation.
- Really huge models do not fit in memory.
- The description of the chain is based on a multidimensional representation of states and events.
- The algorithm needs a description of the macro-states.
- We have considered lumpability but other methods are still under study.

Example: Availability of a multicomponent system



- Processors, Controllers, Disks, 2 types of faults, not independent, not lumpable.
- More than 10^9 states.

Example: Description of the model

- Events are simple or double failure (only the processors), and replacement
- States are the configurations of failed components of each type
- Macro states are the number of failed components (without type)
- Preprocessing: Describe the macro-states
- For each event:
  - Describe the set of initial states
  - Describe the effect of the event
  - Describe the probability of the event

Example: Muntz Availability ctd..

- Main Operation: For each event e and Macro State C_1:
  - Find the macro state C_2 with the largest number which is reached by event e for a state in C_1.
  - Find the maximal probability of event e in C_1.
- Upper bound : when a double fault occur for a state in a macro state it occurs for all the states in the macro-state with the same probability.
- Generation of the lumped matrix (10^6 instead of 10^9 states).

Many applications

- Performance Evaluation
- Reliability (MTTF, point availability)
- Model Checking [7, 25, 41] (but the answer may be "With the bound I am not able to answer True or False") and some operators have to be studied more carefully.
- Qualitative results.
- Proof of convergence.

References

[1] Abu-Amsha O., VincentJ.-M.: An algorithm to bound functionals of Markov chains with large state space. Int: 4th INFORMS Conference on Telecommunications, Boca Raton, Florida, (1998)

[2] Benmammoun M.: Encadrement stochastiques et évaluation de performances des réseaux, PHD, Université de Versailles St-Quentin en Yvelines, (2002)

[3] Benmammoun M., Busic A., Fourneau J.M., and Pekergin N., Increasing convex monotone Markov chains: theory, algorithms and applications, Markov Anniversary Meeting, 2006, Boson Books, pp 189-210.

[4] Benmammoun M., Fourneau J.M., Pekergin N., Troubnikoff A.: An algorithmic and numerical approach to bound the performance of high speed networks, IEEE MASCOTS 2002.

[5] Ben Mamoun M. and Pekergin N.: Closed-form stochastic bounds on the stationary distribution of Markov chains. Probability in the Engineering and Informational Science, 16, pp 403-426, 2002.

[6] Ben Mamoun M., and Pekergin N.: Computing closed-form stochastic bounds on transient distributions of Markov chains. Workshop of Modelling and Performance Evaluation of Quality of Service in Next Generation Internet in SAINT2005, Trento, pp 260-264, IEEE Computer Society 2005.

[7] Ben Mamoun M., Pekergin N., Younès S.: Model Checking of continous-time Markov chains by closed-form bounding distributions. Quantitative Evaluation Sytems QEST2006, Riverside, pages 199-211, IEEE Computer Society 2006.

[8] Buchholz P., An improved method for bounding stationary measures of finite Markov processes, Perform. Eval., V62, N1-4, (2005), pp 349-365.

[9] Busic A.: Comparaison stochastique de modèles Markoviens: une approche algorithmique et ses applications en fiabilité et en évaluation de performances, PHD, Université de Versailles St-Quentin en Yvelines, (2007)

[10] Busic A., Benmammoun M., Fourneau J.M., Modeling Fiber Delay Loops in an All Optical Switch, IEEE QEST 2006, USA.

[11] Busic A., Czarchoski T., Fourneau J.M. Grochla K. , Level Crossing Ordering of Markov Chains: Proving Convergence and Bounding End to End Delays in an All Optical Network, ValueTools 2007, Nantes.

[12] Busic A., Fourneau J.M., A Matrix Pattern Compliant Strong Stochastic Bound, Workshop of Modelling and Performance Evaluation for Quality of Service in Next Generation Internet in IEEE SAINT2005, Trento, (2005), pp 256-259.

[13] Busic A., Fourneau J.M., Bounds for Point and Steady-State Availability: An Algorithmic Approach Based on Lumpability and Stochastic Ordering, European Performance Engeeniring Workshop (EPEW 05), LNCS 3670, Versailles, (2005), pp 94-108.

[14] Busic A., Fourneau J.M., Bounds based on lumpable matrices for partially ordered state space, Structured Markov Chains Tools Workshop in ValueTools 2006, Pisa.

[15] Busic A., Fourneau J.M. and Nott D., Deflection Routing on a Torus is Monotone, Positive System Theory and Application 2006, Springer Verlag LNCIS.

[16] Busic A., Fourneau J.M. and Pekergin N., Worst case analysis of batch arrivals with the increasing convex ordering. European Performance Engeeniring Workshop (EPEW 06), Springer LNCS 4054, pp 196-210, (2006).

[17] Courtois P.J., Semal P.: Bounds for the positive eigenvectors of nonnegative matrices and for their approximations by decomposition. In: Journal of ACM, V 31 (1984) pp 804–825.

[18] Courtois P.J., Semal P.: Computable bounds for conditional steady-state probabilities in large Markov chains and queueing models. In: IEEE JSAC, V4, N6, (1986)

[19] Dayar T., Fourneau J.M., Pekergin N.: Transforming stochastic matrices for stochastic comparison with the st-order, RAIRO-RO V37, pp 85–97, (2003).

[20] Dayar T., Fourneau J.M., Pekergin N. and Vincent J.M., Polynomials of a stochastic matrix and strong stochastic bounds, Markov Anniversary Meeting, (2006), Ed by Boson Books, pp 211-228.

[21] Pekergin N., Dayar T., Alpaslan D., Componentwise bounds for nearly completly decomposable Markov chains using stochastic comparison and reordering. *European Journal of Operational Research*, V165 (2005), pp 810-825.

[22] Dayar T., Pekergin N., Younès S.: Conditional steady-state bounds for a subset of states in Markov chains. Int. Workshop of tools for solving Structured Markov chains, SMCtools06, ACM Press, Pise, Italie, 2006.

[23] Fourneau J.M., Le Coz M., and Quessette F. Algorithms for an irreducible and lumpable strong stochastic bound, Linear Algebra and Applications, Vol 386, 2004, pp 167–186,

[24] Fourneau J.M., Pekergin N.: An algorithmic approach to stochastic bounds Performance Evaluation of Complex Systems : Techniques et Tools (Performance 2002 Tutorial Lecture Notes, pp 64-89, LNCS Springer Verlag 2459, 2002.

[25] Fourneau J.M., Pekergin N., Younès S.: Improving stochastic model checking with Stochatic bounds. distributions of Markov chains. Workshop of Modelling and Performance Evaluation of Quality of Service in Next Generation Internet in SAINT2005, Trento, pp 264-268. IEEE Computer Society (2005).

[26] Fourneau J.M., Plateau B., Sbeity I. and Stewart W.J., SANs and Lumpable Stochastic Bounds: Bounding Availability, in Computer System, Network Performance and Quality of Service, Imperial College Press.

[27] Glasserman P. and Yao. D., Monotone Structure in Discrete-Event Systems. John Wiley & Sons, (1994).

[28] Golubchik, L. and Lui, J., Bounding of performance measures for a threshold-based queuing systems with hysteresis. In: Proceeding of ACM SIGMETRICS'97, (1997) pp 147–157.

[29] Haddad S. and Moreaux S., Sub-stochastic matrix analysis for bounds computations. In: European Jour. of Operational Research, V176 (2007) pp 999–1015.

[30] Hillston J., Kloul L., An Efficient Kronecker Representation for PEPA Models. PAPM'2001, Aachen Germany, (2001).

[31] Keilson J., Kester A., Monotone matrices and monotone Markov processes. Stochastic Processes and Their Applications, V5 (1977) pp 231–241.

[32] Kijima M., Markov Processes for stochastic modeling. Chapman & Hall (1997)

[33] L. Kloul and J.M. Fourneau, A precedence PEPA Model for Performance and reliability analysis. European Performance Engeeniring Workshop (EPEW 06), Springer LNCS 4054, pp 1-15, (2006).

[34] Lui, J. Muntz, R. and Towsley, D.: Bounding the mean response time of the minimum expected delay routing policy: an algorithmic approach. In: IEEE Transactions on Computers. V44 N12 (1995) pp 1371–1382.

[35] Lui, J. Muntz, R. and Towsley, D.: Computing performance bounds of Fork-Join parallel programs under a multiprocessing environment. In: IEEE Transactions on Parallel and Distributed Systems. V9 N3 (1998) pp 295–311.

[36] Massey W.A., Stochastic orderings for Markov processes on partially ordered spaces, Math. Oper. Res., V12, N2, pp 350-367, (1987).

[37] Meyer C.D.: Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. In: SIAM Review. V31 (1989) pp 240–272.

[38] Mosler K. and Scarsini M., Stochastic Orders and Applications: A Classified Bibliography, Springer Verlag Lecture Notes in Economics and Mathematical Systems, V 401, (1993).

[39] Pekergin N.: Stochastic delay bounds on fair queueing algorithms. Proceedings of INFOCOM'99 New York (1999) pp 1212–1220.

[40] Pekergin N.: Stochastic performance bounds by state reduction. Performance Evaluation V36-37 (1999) pp 1–17.

[41] Pekergin N., Younès S.: Stochastic Model Checking with Stochastic Comparison. 2nd European Performance Engineering Workshop 2005, EPEW05, pp 109-124. LNCS 3670, Formal Techniques for Computer Systems and Business Processes (2005).

[42] Shaked M., Shantikumar J.G.: Stochastic Orders and Their Applications. In: Academic Press, California (1994).

[43] Stewart W. J.: Introduction to the Numerical Solution of Markov Chains. Princeton University Press, (1994).

[44] Stoyan D.: Comparison Methods for Queues and Other Stochastic Models. John Wiley & Sons, Berlin, Germany, (1983).

[45] Muller A. and Stoyan D., Comparison Methods for Stochastic Models and Risks, Wiley, (2002).

[46] Trémolière M., Vincent J.M., and Plateau B., Determination of the Optimal Stochastic Upper Bound of a Markovian Generator, ORSA-TIMS Conference, (1993).

[47] Truffet L. : Near Complete Decomposability: Bounding the error by a Stochastic Comparison Method. Advances in App. Prob. Vol.29, (1997) 830-855.

[48] Truffet L.: Reduction Technique For Discrete Time Markov Chains on Totally Ordered State Space Using Stochastic Comparisons. In: Journal of Applied Probability, V37 N3 (2000).

[49] Van Dijk N.: Error bound analysis for queueing networks", In: Performance 96 Tutorials, Lausanne, (1996).