Bounding transient and steady-state dependability measures through algorithmic stochastic comparison

Ana Bušić and Jean-Michel Fourneau [1]

1 Introduction

We are interested in bounding dependability measures like point and steady-state availability and reliability of systems modelled by very large Markov chains which are not numerically tractable. We suppose that the state space is divided into two classes, UP (system is operational) and DOWN states. The reliability at time t is defined as the probability that the system has always been operational between 0 and t. The point availability is the probability that the system is operational at time t, and the steady-state availability is the limit, if it exists, of this probability.

The usual way to compute dependability measures of continuous-time Markov chains (CTMC) is based on uniformization method. Thus we compute bounds on discrete-time Markov chains (DTMC).

2 Algorithmic stochastic comparison approach

Our approach is based on stochastic comparison techniques [5]. We use strong stochastic order ($\leq_{st}$) which is defined through comparison of expectations of increasing functions. Thus this order can be used to compare increasing rewards of two Markov models. The main result we use in our approach is the theorem on the comparison of two DTMC's [5]: if we have two DTMC's X and Y which transitions matrices and initial distributions are $\leq_{st}$-comparable and at least one of the transition matrices is $\leq_{st}$-monotone, then we can compare X and Y at each instant t and, if they exist, the steady-state distributions are comparable too. Moreover, the $\leq_{st}$ comparison and monotonicity can be characterized by linear algebraic constraints allowing thus an algorithmic approach. For the transition matrix P of a given DTMC X, we compute a bounding matrix Q such that $P \preceq_{st} Q$ (relation 1) and the matrix Q is $\preceq_{st}$-monotone (relation 2),

$$\sum_{k=j}^{n} P_{i,k} \leq \sum_{k=j}^{n} Q_{i,k},\ \forall i, j, \tag{1}$$

$$\sum_{k=j}^{n} Q_{i-1,k} \leq \sum_{k=j}^{n} Q_{i,k},\ \forall j,\ \forall i \geq 2. \tag{2}$$

In order to reduce the state-space size of the bounding chain, we force matrix Q to be lumpable, i.e. such that, for a given partition $C_l, l \in L$ of the state space,

$$\forall l \in L, \forall j \in L \sum_{k \in C_j}^{n} Q_{i,k} \text{ is constant } \forall i \in C_l. \tag{3}$$

In [3] an algorithm computing a lumpable irreducible monotone upper bounding Markov chain (LIMSUB algorithm) has been developed. We extended this approach in two directions.

Bounding transient rewards: To our knowledge, the algorithmic stochastic comparison approach has only been used in steady-state analysis. Although the same theory can be used to compute the bounds for both transient and steady-state rewards, there are some points on which those two problems differ and the algorithmic approach must be, therefore, slightly modified. LIMSUB [3] algorithm includes irreducibility constraints as they assure the existence of the steady-state distribution of a bounding (finite and aperiodic) Markov chain. More precisely, LIMSUB algorithm does not allow to remove transitions in the upper triangle of the stochastic matrix and it adds new entries into the sub diagonal. It is shown in [3] that these two constraints imply that the bounding matrix is irreducible. When computing the transient bounds, we are not always interested in having an irreducible bounding chain. Moreover, we might want to analyse the original chains that are not necessarily irreducible. When computing the reliability bounds, for instance, we take into consideration the chain where all the DOWN states are aggregated into one absorbing state. In that case, LIMSUB algorithm cannot be used.

We have adapted the algorithm to compute the bounds of reducible Markov chains. This new algorithm is called LMSUB (Lumpable Monotone Stochastic Bound) and it is based only on relations (1), (2) and (3). It uses a sparse matrix representation of P and Q and only three vectors are in memory. Unlike LIMSUB, LMSUB does not use additional constraints to insure irreducibility. Thus the new algorithm is much simpler.

Avoiding generation of the state space: We have also developed a new algorithm that, using a high level formalism, provides directly a stochastic matrix that is lumpable and $\leq_{st}$-larger, called LL algorithm [1]. Only the lumped matrix is generated, thus we gain significantly both in terms of time and storage complexity. Indeed, for some reliability models, even the state space is too large to fit in memory. LL is based on relations (1) and (3) and may depend on the high level formalism used for the model specifications. Indeed, we do

[1]PRiSM, Université de Versailles St-Quentin, 45, Av. des Etats-Unis 78000 Versailles, France. Email: {abusic,jmf}@prism.uvsq.fr

for $b = m$ **downto** 1 **do**
  for $j = last(b)$ **downto** $first(b)$ **do**
    for $a = 1$ **to** m **do**
      for $i = first(a)$ **to** $last(a)$ **do**
        $sumq_{i,j} = \max(sumq_{i-1,j}, \sum_{k=j}^{n} p_{i,k});$
        if $b < m$ ***and*** $j = last(b)$ ***and*** $i = first(a)$ **then**
        $sumq_{i,j} = \max(r_{a,b+1}, sumq_{i,j});$
      end
    end
  end
  for $a = 1$ **to** m **do** $r_{a,b} = sumq_{last(a),first(b)};$
end

Algorithm 1: LMSUB. P, Q and R denote respectively the initial, lumpable and lumped matrix. $sumq_{i,j} = \sum_{k=j}^{n} q_{i,k}$ with $sumq_{-1,j} = 0$. We consider a partition into m macro states. $first(l)$ and $last(l)$ denote the first and the last state of a macro state. Only the lumped matrix R is actually computed and stored.

not require at the first step to bound a monotone matrix. In the second step, we use LMSUB to obtain a monotone upper bound of the matrix given by LL.

3 Application on repairable systems

We illustrate our approach on the example of repairable multicomponent systems. The typical model consists of several types of components and several units of each component type. The units can fail and the failed units can be repaired. The set of operational (UP) states of the system is defined by a function on the state-space, usually through a minimal subset of working components of each type.

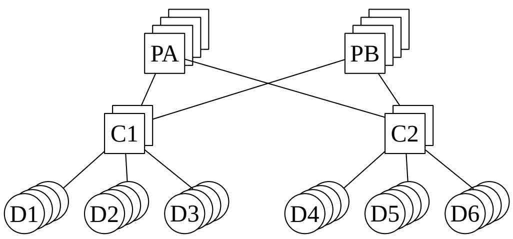

Figure 1: Repairable system example [4]

We have taken the example presented in [4] (Figure 1) and applied our method to derive bounds for point and steady-state availability and for reliability of that system, using the same set of parameters considered in [2] for steady-state analysis. The numerical results for transient dependability measures (point availability and reliability) are given in Figure 2. The lower bound for steady-state availability obtained by our method is 0.999132158.

For more detailed description of the model we refer the reader to [4], [2]. More details on the total ordering used on the state-space and practical remarks on LL algorithm (used to derive point availability bounds) can be found in [1]. Let us

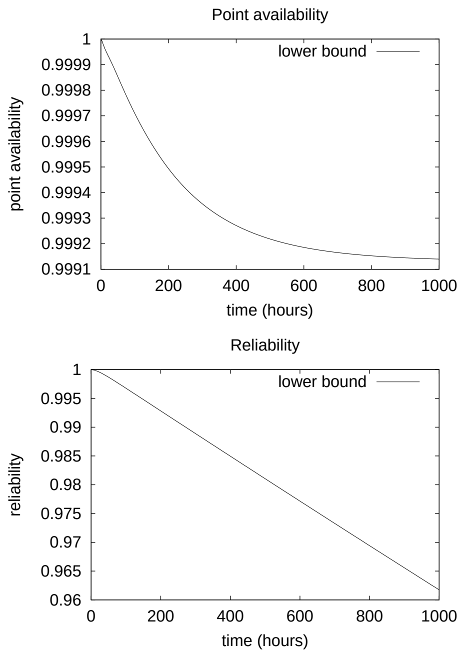

Figure 2: Lower bounds for point availability and reliability for the system on Figure 1

just remark that the original model for analysed system has only 36 components of 10 different types, yet the state-space is of order of 9×10^{10} (more than 10^{12} transitions), which completely excludes direct generation of the state-space. The intermediate model, generated by means of LL algorithm, has only 1312235 states (all the UP states have been generated).

Although steady-state availability bounds seem to be less accurate than those obtained through former methods [4], [2] (both perform only steady-state analysis), we can handle a larger class of problems as we use no special properties on the transition graph structure.

References

[1] Busic A., Fourneau J.-M.: Bounds for Point and Steady-State Availability: an Algorithmic Approach Based on Lumpability and Stochastic Ordering. EPEW 2005, Springer LNCS 3670 (2005), 94–108

[2] Carrasco J.A.: Bounding steady-state availability models with group repair and phase type repair distributions, Performance Evaluation, V 35 (1999), 193–204

[3] Fourneau J.-M., Lecoz M., Quessette F.: Algorithms for an irreducible and lumpable strong stochastic bound. Linear Algebra and its Applications, V 386 (2004) 167–185

[4] Muntz R., de Souza e Silva E., Goyal A.: Bounding availability of repairable computer systems. IEEE Trans. on Computers, V 38 (12) (1989), 1714–1723

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