Applications of Mathematics

Štěpán Klapka; Petr Mayer

Aggregation/disaggregation method for safety models

Applications of Mathematics, Vol. 47 (2002), No. 2, 127–137

Persistent URL: http://dml.cz/dmlcz/134490

47 (2002) APPLICATIONS OF MATHEMATICS No. 2, 127–137

AGGREGATION/DISAGGREGATION METHOD FOR SAFETY MODELS*

Štěpán Klapka, Petr Mayer, Praha

Abstract. The paper concerns the possibilities for mathematical modelling of safety related systems (equipment oriented on safety). Some mathematical models have been required by the present European Standards for the railway transport. We are interested in the possibility of using Markov's models to meet these Standards. In the text an example of using that method in the interlocking equipment life cycle is given. An efficient aggregation/disaggregation method for computing some characteristics of Markov chains is presented.

Keywords: Markov chain, stochastic matrix, stationary probability vector, aggregation/disaggregation algorithms

MSC 2000: 65F10, 65F15, 15A51

1. Introduction

The life cycle of software (SW) is usually divided into several phases. Such phases are specification, implementation, integration, verification and testing, and maintenance. The costs of the specification error correction are positively related to the time of detection. To decrease the development costs it is essential to verify the specification or the SW design before or at the beginning of the implementation phase. Sometimes this verification changes the requirements needed by the system, on hardware architecture (HW). Generally, formal methods, which use mathematical theories to analyse the specification process, are applied to solve problems related to the changes made in the requirements. Following this, analysis and research are carried out, the results of which are used in the subsequent phases of SW development.

PSM (Probabilistic Structured Modelling) is used to test certain new conceptions in AZD Praha Company Ltd. This model originates from a finite state machine,

* This work was partly supported by the Grant No. 201/98/0528 of the Grant Agency of the Czech Republic and the Grant MSM J13/98:113200007.

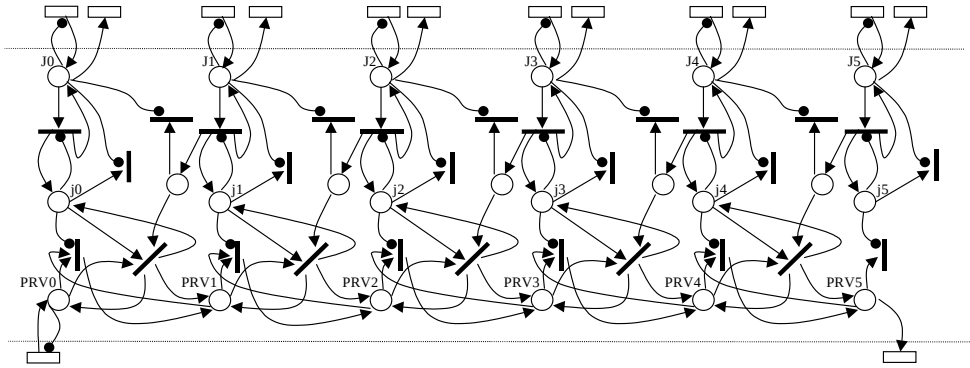


Figure 1. The GSPN model of PRV for ABE-1.

which is determined by logical connections and by probability models of faults. For the construction of the final model, the Stochastic Automata Network (SAN) or the Stochastic Petri Nets (SPN), see Fig. 1, are used. The tendency for application of quantitative risk analysis suggested by the European Standards is EN 50 159-1, EN 50 126, EN 50 129, EN 50 128. The notion of "safety" is defined as the probability that faults are either non-existent or exist but are detected. A detailed explanation regarding this probability may be found in [3] together with the Markov simple absorption model for safety. The analytical solution of simple absorption models for several configurations of critical application are mentioned in [10]. Approaching the Markov models in real life increases the number of states. This procedure is known as the state space explosion. It is very difficult to obtain an analytical or numerical solution for such a huge model. If it is modelled by ergodic DTMC (Discrete Time Markov Chain), it is possible to use some of the steady state solution technology.

2. Markov Models

Before passing on to the Markov model, it is important to note that most of the analytical utilities such as several kinds of Stochastic Petri Nets (SRN, GSPN, QPN) use some form of Markov modelling. These are explained in greater detail in references [1], [2], [8]. The basic terminology and methodology for modelling by a Markov chain is also available in [11], [12]. A stochastic process is a set of random variables $\{X(t);\ t \in T\}$ which are defined on a probability space, indexed by a parameter t, where t is a variable from the parameter space T. The variable $X(t)$ is an observation at time t. If T is discrete, it is referred to as a discrete time stochastic process. Conversely, if T is continuous the process is known as a continuous time stochastic process. A stochastic process is stationary if it is invariant with respect

to time delay. The value that the variable $X(t)$ reaches is known as the state. The set of states, that is the state space, can be discrete or continuous.

The Markov process is a stochastic process which satisfies the memoryless condition known as the Markov property, as shown below:

$$\begin{aligned}(1)\qquad \text{Prob}\{X(t) \leqslant x \mid X(t_0) = x_0, \ldots, X(t_n) = x_n\} \\ = \text{Prob}\{X(t) \leqslant x \mid X(t_n) = x_n\}.\end{aligned}$$

If the departure of the state $X(t)$ is dependent on the parameter t, the Markov process is referred to as nonhomogeneous. If not, it is referred to as homogeneous. If the state space of the Markov process is discrete, it is called a Markov chain. In this case a subset of natural numbers is chosen.

Two types of Markov chains may be identified. These are discrete time DTMC and continuous time CTMC. When the time is continuous (CTMC) the Markov property causes an exponential distribution of the time which is spent in a fixed state. When the time is discrete the Markov property causes a geometrical distribution. As an example of CTMC, one may see a simple model for the life cycle of a safe critical system (see Fig. 2). This system can be described in the following four different stages and transition rates. These are:

1. Faultless state.
2. State when a detectable fault exists.
3. State when a second fault has occurred, which makes it impossible for the first to be detected. This is considered as a highly dangerous state.
4. The repairing stage. It is risky if you are not equipped with resources which need to be replaced.

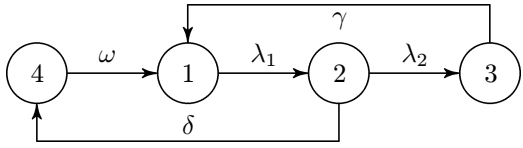


Figure 2. Life cycle example.

In Fig. 2, λ_1 is the rate of simple detectable faults, λ_2 is the rate of the second fault, δ is the rate of the fault detection, γ is the rate of how often the system is replaced, ω is the rate of repair. The following infinitesimal generator Q describes Homogenous Continuous Time Markov Chain:

$$(2)\qquad Q = \begin{bmatrix} -\lambda_1 & \lambda_1 & 0 & 0 \\ 0 & -\delta - \lambda_2 & \lambda_2 & \delta \\ \gamma & 0 & -\gamma & 0 \\ \omega & 0 & 0 & -\omega \end{bmatrix}$$

When the replacement of a system is not essential, then γ and ω are equal to zero and the states 3 and 4 are Absorption States. It is impossible to get away from an Absorption State. The system persists in a fixed state so the probability is one. The following system of ordinary differential equations with constant coefficients describes the changes of the equipment at a time t, where u is the vector of probability that the equipment is in a given state at time t:

$$\frac{\mathrm{d}u^T}{\mathrm{d}t} = Qu^T, \quad u(t) = \big(u_1(t), \ldots, u_n(t)\big). \tag{3}$$

For electronical equipment which works in time steps, it is natural to use the DTMC model. The purpose of this model is to verify whether the algorithm solves fault situations. The usual situations are an I/O subsystem, bus system, communication subsystem, and protocols.

DTMC are established by a transient matrix $P(k) = (p(k)_{ij}) \in \mathbb{R}^{N\times N}$ which describes one step transition probabilities in step k. The elements of the matrix are the probabilities of transition from one state to another in the k-th time step:

$$p(k)_{ij} = \mathrm{Prob}\{X_{k+1} = j \mid X_k = i\}, \tag{4}$$

while the probability that the process in the n-th step goes from the state i to the state j is defined by the matrix

$$P(m)^{(n)} = (p(m)_{ij}^{(n)}),$$
$$p(m)_{ij}^{(n)} = \mathrm{Prob}\{X_{m+n} = j \mid X_m = i\}.$$

For $P(m)^{(n)}$ we can write

$$P(m)^{(n)} = P(m)P(m+1)\ldots P(m+n-1).$$

For any l the Chapman-Kolmogorov equation holds:

$$P(m)^{(n)} = P(m)^{(l)}P(m+l)^{(n-l)}.$$

In the case of homogenous DTMC, this means that when one step transition matrices do not depend on the step number, i.e. $P(k) = P(l)$ for any k, l, we can simplify the above formulas. We can write

$$P^{(n)} = P^n.$$

For the description of the process we can use a nonnegative vector u with the sum of all elements equal to one. Each element $u_i^{(k)}$ of the vector describes the probability

for the Markov chain to be in the state i in the step k. The formula below shows us how we can calculate the probability vector u_{n+m} at the time $n+m$ from the known vector u_n at the time n:

$$u^{(n+m)^T} = u^{(n)^T} P^m.$$

One very important information is the solution of the equation

$$u^T = u^T P, \quad u^T e = 1, \tag{5}$$

where e is the vector of all ones. The solution of (5) can be interpreted as the long term behaviour of the Markov chain, usually it is called the *stationary probability vector*.

R e m a r k. If for solving problem (3) we use the explicit Euler method with a small enough time step τ we get

$$u(t+\tau) = u(t) + \tau Q u(t) = (I + \tau Q) u(t)$$

where the matrix $I + \tau Q$ is stochastic. With a fixed time step we can analyse problem (3) as a DTMC instead of CTMC.

3. Aggregation/disaggregation algorithm

Now we address some effective methods for computing stationary probability vectors. We will analyze aggregation/disaggregation methods which show very good behaviour especially in the case of large Markov chains. It is well known from literature ([6], [11]) that such type of methods is extremely good in the case of irreducible, but nearly completely decomposable matrices. Nonetheless, even if the matrix is only irreducible, the method is very useful.

Now we shall define a mapping $g = 1, \ldots, N \longmapsto 1, \ldots, n$ which maps the indices to aggregation groups. Using g we define communication operators, the restriction R, and for $x > 0$ the prolongation operator $S(x)$

$$(Rx)_i = \sum_{g(j)=i} x_j, \quad x \in \mathbb{R}^N,$$

$$(S(x)z)_i = z_{g(i)} \frac{x_i}{(Rx)_{g(i)}}.$$

For nonzero elements of R and $S(x)$ we can write

$$R_{g(j)j} = 1, \tag{6}$$

$$S(x)_{ig(i)} = \frac{x_i}{(Rx)_{g(i)}}. \tag{7}$$

Algorithm 1. SPV (P, T, Y, s, t)

Let P be an irreducible stochastic matrix, let g define the aggregation, let $\widetilde{x}_1 > 0$ be the initial vector.

Let $\widehat{P} = (I + P^T)/2$, $A = I - \widehat{P} = M - K$. Define $T = (M^{-1}K)^t$ and $Y = (\widehat{P})^s$ for $t \geqslant 1$ and $s \geqslant 1$, $x_1 = T\widetilde{x}_1$. Let $\varepsilon > 0$ be the final tolerance.

Step 1. Set $k = 1$

Step 2. Construct

$$B(x_k) = RYS(x_k) \tag{8}$$

Step 3. Solve $B(x_k)z_k = z_k, \;\; e^T z_k = 1$

Step 4. Set $v_{k+1} = S(x_k)z_k$

Step 5. Compute $x_{k+1} = Tv_{k+1}$

Step 6. If $\|x_k - x_{k+1}\| \geqslant \varepsilon$ then $k = k + 1$, go to Step 2

Step 7. Stop

One can think about the SPV Algorithm 1 as a kind of a two grid method. The steps 2, 3, 4 are equivalent to the coarse grid correction, step 5 is a smoothing, with the operator T as the smoother.

The theorem below, which is proved in [6], states the local convergence of the algorithm SPV.

Theorem 1. *Let P be an irreducible stochastic matrix, let g be a mapping which defines the aggregation, let M, K be splittings defining the smoother T and the matrix Y, respectively. Then Algorithm* 1 *is locally convergent.*

We will be interested mainly in three types of smoothers:

1. Simple power method $M = I, \;\; K = \widehat{P}$.
2. Block Jacobi method—known as Vantilborg method with $M_{ij} = A_{ij}$ when $g(i) = g(j)$.
3. Block Gauss-Seidel mehod—known as Koury-McAllister-Stewart method with $M_{ij} = A_{ij}$ when $g(i) \geqslant g(j)$.

The first of them is, of course, the slowest. But in many situations one may be unable to get the matrix P elementwise. For example, when we analyze Stochastic Automata Networks, P is formed as a sum of tensor products. In such situations, the first smoother can be the only one which is computable.

4. Special situation for the choice of aggregation groups

In many situations the transition matrix P has a special form. Now we study the convergence behaviour of the SPV algorithm.

Lemma 1. *Let $P \in \mathbb{R}^{N\times N}$ be a stochastic matrix, let g be a mapping which defines the aggregation. For vectors $x, y \in \mathbb{R}^N$ let there exist nonzero values α_k, $k = 1, \ldots, n$, such that $y_i = \alpha_{g(i)} x_i$. Then $S(x) = S(y)$ and $B(x) = B(y)$.*

P r o o f. Due to (8) we need only to show that $S(x) = S(y)$. Because the nonzero structure of both matrices is the same, we need to check only the nonzero elements. From (6) and (7) we get

$$S(y)_{ig(i)} = \frac{y_i}{(Ry)_{g(i)}} = \frac{\alpha_{g(i)} x_i}{\sum_{j,g(j)=g(i)} y_j} = \frac{\alpha_{g(i)} x_i}{\sum_{j,g(j)=g(i)} \alpha_{g(i)} x_j}$$
$$= \frac{\alpha_{g(i)} x_i}{\alpha_{g(i)} \sum_{j,g(j)=g(i)} x_j} = \frac{x_i}{(Rx)_{g(i)}} = S(x)_{ig(i)}.$$

□

From Lemma 1 we have

Corollary 1. *Let P be an irreducible stochastic matrix. Let g be the mapping defining the aggregation. Let u^* be a solution of the problem* (5)*, for the vector u let there exist nonzero values α_k such that $u_i = \alpha_{g(i)} u_i^*$. Then the coarse correction (i.e. steps* 2, 3, 4 *in Algorithm* 1*) computes the exact solution starting with $x_1 = u$.*

P r o o f. From Lemma 1 we know that $S(u) = S(u^*)$ and $B(u) = B(u^*)$. Then z computed in Step 3 is the same for u and u^*. For the same reason we have $S(u)z = S(u^*)z = u^*$. □

Now we can show some special types of matrices for which we get very fast convergence, or which even converge in one step. A different proof of the next theorem can be found in [7].

Theorem 2. *Let g define the aggregation, let there exist vectors $v^{(i)}, w^{(i)} \in \mathbb{R}^N$ and a matrix C such that $P = \sum_{i=1}^{n} w^{(i)} {v^{(i)}}^T + C$ is an irreducible stochastic matrix, $C = (c_{ij})$, $c_{ij} = 0$ when $g(i) \neq g(j)$, $v_j^{(i)} = 0$ when $g(j) \neq i$, $w_j^{(i)} = 0$ when $g(j) = i$. Let $\widetilde{x}_1$ be not orthogonal to any vector $w^{(i)}$. Then Algorithm* 1 *with the Jacobi or Gauss-Seidel smoother constructs the exact solution after the first iteration.*

P r o o f. All we need to show is that $x_1 = T\widetilde{x}_1$ satisfies the conditions from Corollary 1. We have

$$A = \frac{1}{2}(I - P^T) = \frac{1}{2}(I - C) - \frac{1}{2}\sum_{i=1}^{n} v^{(i)} w^{(i)^T}.$$

Now we can analyse the Jacobi method as a smoother. In this situation we have

$$M = \frac{1}{2}(I - C), \quad K = \frac{1}{2}\sum_{i=1}^{n} v^{(i)} w^{(i)^T},$$

hence

$$T = (I - C)^{-1} \sum_{i=1}^{n} v^{(i)} w^{(i)^T}.$$

By the definition of x_1 we have

$$x_1 = T\widetilde{x}_1 = (I - C)^{-1} \sum_{i=1}^{n} v^{(i)} w^{(i)^T} \widetilde{x}_1 = \sum_{i=1}^{n} \beta_i \widetilde{v}^{(i)}$$

where $\beta_i = w^{(i)^T} \widetilde{x}_1 \neq 0$ and $\widetilde{v}^{(i)} = (I - C)^{-1} v^{(i)}$. Since the exact solution u^* satisfies $u^* = Tu^*$ we have $u^* = \sum_{i=1}^{n} \beta_i^* \widetilde{v}^{(i)}$. Recall that c_{ij} could be nonzero only in the case $g(i) = g(j)$ and the same is true for the matrix $(I-C)^{-1}$. Similarly, nonzero elements $v_j^{(i)}$ are only possible when $g(j) = i$. Then all nonzero elements $\widetilde{v_j}^{(i)}$ are in places where $g(j) = i$. Hence we can write

$$(x_1)_i = \frac{\beta^*_{g(i)}}{\beta_{g(i)}} u_i^*$$

and use Corollary 1 to complete the proof for the Jacobi smoother. Nearly the same arguments can be used to prove the Gauss-Seidel case. □

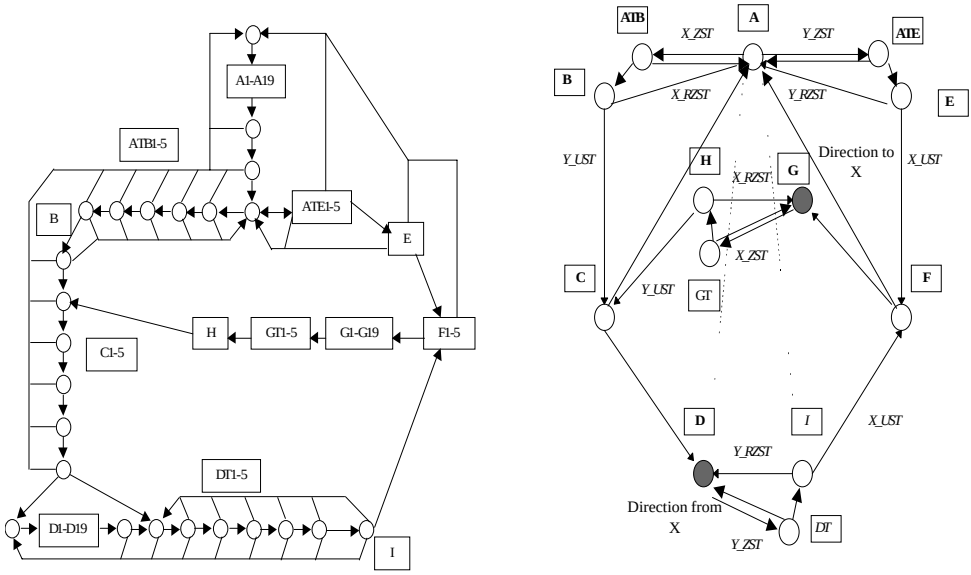
5. Example

In special DTMC cases, the convergence for standard numerical methods is very slow. The problem is not necessarily so large. For this situation the aggregation/disaggregation iterative algorithm is useful. This is explained in greater detail in references [6], [4]. An example of the problematic size of the model can be the finite state machine for controlling the line block direction ABE-1 (see Fig. 3b). The most important safety principle for changing the direction of the line block are the

protected timeouts. During these timeouts it is impossible to change important signals. During this period the occupation of the track is checked, which is the condition for making the direction change possible. These states are A, C, D, G, E, F, which correspond to sequences A1–A19, C1–C5 ... The transitions which require the stable state without external changes are linked to states ATB, ATE, GT, DT. For such a system, a very fast iteration is proved provided the block iteration method together with the aggregation type is used. Further details can be found in reference [4]. When the right aggregation group of states is chosen, the procedure converges to one step of iteration. For the matrix for the Markov chain described in Fig. 3 we have the following number of iterations:

Smoother	Smoother only (spectral radius)	Aggregation with smoother (to get residual less than 1e-14)
power method	0.9999999999928	10
Jacobi method	0.9999999927499	1
G.S. method	0.9999999999599	1

Table 1.



(a) Simplified graph of transition matrix (b) State diagram

Figure 3. Description of state automata for line block directional control.

Huge DTMC models produce a net of finite state machines which are synchronised at special transitions. Without synchronisation, the transition matrix of the model is

the tensor product of all considered transition matrices in the net. Synchronisation events determine that the final matrix is the sum of tensor products. This is useful for the modelling of the overloading CPU interrupt system. This technology describes how to fight against state space explosion. This is explained in greater detail in reference [9].

6. Conclusion

We have analysed some of possible iteration solvers for the analysis of large Markov chains. We have shown that there are special situations when convergence can be achieved in one iteration. Further analysis of possible methods for reducible Markov chains is necessary. Another problem, which is not yet solved satisfactorily, is the situation when only an approximate solution of coarse level correction is available. Such type of analysis can lead to a real multilevel algorithm for computing the stationary probability vector, instead of the two level method.

Next, a large set of problems is to compute not only the stationary probability vector but the mean first passage times and their variations. In this situation the reducible case is of special interest.

References

[1] *G. Ciardo, A. Blakmore, P. F. Chimento, JR., J. K. Muppala and K. S. Trivedi*: Automated generation and analysis of Markov reward models using stochastic reward nets. In: Linear Algebra, Markov Chain, and Queueing Models (C. D. Meyer, R. J. Plemmons, eds.). Springer-Verlag, New York, 1993, pp. 145–191.

[2] *R. David, H. Alla*: Petri Nets and Grafcet: Tools for Modelling Discrete Event Systems. Prentice Hall International, 1992.

[3] *B. W. Johnson*: Design and Analysis of Fault-Tolerant Digital Systems. Addison-Wesley Publishing Company, Massachusetts, 1989.

[4] *Š. Klapka, P. Mayer*: Some aspects of modelling railway safety. In: Proceedings of the XIIIth SANM, Nečtiny. Západočeská univerzita, Plzeň, 1999, pp. 135–140.

[5] *K. Kule*: Reliability and safety of interlocking systems. NADAS, Praha, 1980. (In Czech.)

[6] *I. Marek, P. Mayer*: Convergence analysis of an iterative aggregation/disaggregation method for computing stationary probability vectors of stochastic matrices. Numer. Linear Algebra Appl. *5* (1998), 253–274.

[7] *I. Marek, P. Mayer*: Iterative aggregation/disaggregation methods for computing stationary probability vectors of stochastic matrices can be finitely terminating. J. Differential Equations *3* (2001), 301–313.

[8] *M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis*: Modelling with Generalized Stochastic Petri Nets. John Wiley & Sons, Chichester, 1995.

[9] *B. Plateau, K. Atif*: Stochastic automata network for modelling parallel systems. IEEE transaction on software engineering *17* (1991), 1093–1108.

[10] *K. Rástočný*: Models for analysis of safety computer interlocking systems. Habilitation thesis. University of Žilina, 1998. (In Slovak.)

[11] *W. J. Stewart*: Introduction to the Numerical Solution of Markov Chains. Princeton University Press, Princenton, 1994.
[12] *J. Walter*: Stochastic Models in Economy. SNTL, Praha, 1970. (In Czech.)

Authors' addresses: *Štěpán Klapka*, AŽD Praha Ltd., Závod technika, Research and Development Department, Žírovnická 5, 106 17 Praha 10, Czech Republic, e-mail: `klapka.stepan.vav@azd.cz`; *Petr Mayer*, Institute of Numerical Mathematics, Faculty of Mathematics and Physics, Charles University, Sokolovská 83, 186 75 Praha 8, Czech Republic, e-mail: `pmayer@ms.mff.cuni.cz`.