UNAVAILABILITY CALCULATIONS WITHIN THE LIMITS OF COMPUTER ACCURACY

R. Briš
•
Technical University of Ostrava, Faculty of Electrical Engineering and Computer Science, Ostrava, Czech Republic

e-mail: radim.bris@vsb.cz

ABSTRACT

The paper presents a new analytical algorithm which is able to carry out direct and exact reliability quantification of highly reliable systems with maintenance (both preventive and corrective). A directed acyclic graph is used as a system representation. The algorithm is based on a special new procedure which permits only summarization between two or more non-negative numbers that can be very different. If the summarization of very small positive numbers transformed into the machine code is performed effectively no error is committed at the operation. Reliability quantification is demonstrated on a real system from practice.

1 INTRODUCTION

It is a simulation method which is used for the quantification of reliability when accurate analytic or numerical procedures do not lead to satisfactory computations of system reliability. A direct simulation technique has been improved by the application of a parallel algorithm [1] to such extent that it can be used for real complex systems which can be modelled and quantitatively estimated from the point of view of the reliability without unreal simplified conditions which analytic methods usually require. However, if it is necessary to work and quantitatively estimate highly reliable systems, when unreliability indicators (i.e. system non-functions) move in the order 10^{-5} and higher (i.e.10^{-6} etc.), the simulation technique, whatever improved, can meet the problems of prolonged and inaccurate computations.

Highly reliable systems appear more often and in research they are closely connected with a penetrative increase of progress. We can observe the systems for example in space applications where complex device is often designed for a few tens of years without a possibility of help of human hand. Safety systems of nuclear power stations represent other example of highly reliable systems. They have to be reliable enough to comply with still increasing internationally agreed safety criteria and moreover they are mostly so called sleeping systems which start and operate only in the case of big accidents. Their hypothetical failures are then not apparent and thus reparable only at optimally selected inspective times. We can add some more examples. The question is how to model the behaviour of these systems and how to find their efficient procedure for estimation and computation of reliability indicators.

2 A PROBLEM FORMULATION AND COMPONENT MODELS

Let us have a system assigned with a help of a directed acyclic graph (AG) [1]. Terminal nodes of the AG that represent functionality of input system components are established by the definition of deterministic or stochastic process, to which they are subordinate. From them we can compute a time course of the availability coefficient, possibly unavailability of individual terminal nodes, using methodology of basic renewal theory, as for example in [2]. The aim is then to find a correspondent time course of the unavailability coefficient for the highest SS node which represents reliability behaviour of the whole system.

2.1 Models of components – terminal nodes

In the first phase of the research, an exponential distribution for the time to a failure will be supposed, possibly for the time to a restoration. Under this condition, all frequently used models with both preventive and corrective maintenance may be described by three of the following models:

- Model with elements (terminal nodes in AG) that can not be repaired
- Model with repairable elements (CM – Corrective Maintenance) for apparent failures, i.e. a
  model when a possible failure is identified at the occurrence and immediately afterwards it starts a process leading to its restoration.
- Model with repairable elements with hidden failures, i.e. a model when a failure is identified only at special deterministically assigned times, appearing with a given period (moments of periodical inspections). In the case of its occurrence at these times an analogical restoration process starts, as in the previous case.

An analytical accurate computation of time dependence of the (un)availability coefficient was for the first two situations explained enough and derived in [2]. Let us remind that in the first case of the element that can not be repaired a final course of unavailability coefficient *P(t)* is presented by a distribution function of time to failure of the element:

$$P(t) = 1 - e^{-\lambda t}, \tag{1}$$

where λ is the failure rate.

In the second case we can derive a relation on the basis of Laplace´s transformation for a similar coefficient

$$\begin{aligned} P(t) &= 1 - \left[\frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)t} \right] \\ &= \frac{\lambda}{\lambda+\mu}\left[1 - e^{-(\lambda+\mu)t}\right], \quad t > 0 \end{aligned} \tag{2}$$

where μ is the repair rate.

The third model is rather more general than the earlier stated in the model with periodical exchanges [2] where we assumed a deterministic time to repair. If we hereby presume that a time to the end of repair is exponential random variable, it is necessary to derive an analytical computation of time course of the function of unavailability coefficient.

2.2 Unavailability coefficient for a model with repairable elements and hidden failures

With the same indication of failure and restoration intensities as given above we can describe the unavailability coefficient with the following function:

$$P(\tau) = (1 - P_C).(1 - e^{-\lambda\tau}) \\ + P_C \left[1 + \frac{\mu}{\mu - \lambda} (e^{-\mu\tau} - e^{-\lambda\tau}) \right], \quad \tau > 0 \tag{3}$$

where *τ* is a time which has passed since the last planned inspection, *Pc* is the probability of a non-functional state of an element at the moment of inspection at the beginning of the interval to the next inspection.

Proof of the relationship (3) brings ref. [3].

Note:

1. For the purposes of an effective computer calculation the expression in the brackets can be converted into the formation:

$$\left[1 + \frac{\mu}{\mu - \lambda} (e^{-\mu\tau} - e^{-\lambda\tau}) \right] = \\ 1 - \frac{\mu}{\mu - \lambda} e^{-\lambda\tau} \left[1 - e^{-(\mu - \lambda)\tau} \right], \quad \tau > 0 \tag{4}$$

2. In other hypotheses we will need this expression to be always positive, what is also easy to proof.

3 THE NEW ALGORITHM

3.1 Probabilities of functional and non-functional state

It is evident that probabilities of a functional *p* and non-functional state *q* comply with a relation

$q + p = 1.$

Taking into consideration the final accuracy of real numbers in a computer it is important which one from *p* or *q* we count. If we want to fully use the machine accuracy, we have to compute the smaller one from both probabilities.

Example:

We take into account the following sum:

0.000 002 7816 + 0.999 997 2184

If we counted hypothetically on a computer with three-digit decimal numbers, then for the value of *q* = 0.00000278, we would instead of a correct value *p* = 0.9999972184 have only *p* = 1**.**

In return for *q* = 1- *p*, we would get: *q* = 1- *p* = 0, keeping at disposal *p* = 1.

It is apparent that it gets to a great loss of accuracy if we firstly counted *p* instead of *q*. Our result will be maximally precise saving accuracy of *q*.

Seeing that probabilities of a non-function state of a highly reliable system is very small, we have to concentrate on numerical expression of these probabilities. For these purposes it is necessary to reorganize the computer calculation and set certain rules which do not have the influence on accuracy of the computation at the numeration process.

3.2 Probability calculation of non-functional states of terminal nodes

The probability calculation of non-functioning state (unavailability coefficient) of the simplest possible not repaired element (or terminal node) can be done by the use of relation (1).

Similarly, for other models of system elements the computation of an expression

$$1-e^{-x}, \tag{5}$$

for x ≥ 0 is a crucial moment at the probability numerical expression of a non-function state (unavailability coefficient).

For values x<<1, i.e. near 0, direct numerical expression written by the formula would lead to great errors! At subtraction of two near numbers it gets to a considerable loss of accuracy. On personal computer the smallest number ε, for which it is numerically evident that

$$1+\varepsilon \neq 1,$$

is approximately 10^{-18}.

If

$$x \approx 10^{-25},$$

the real value of the expression (5) will be near 10^{-25}. A direct numerical calculation of the expression gets a zero!

As the algorithm was created in a programming environment Matlab, for the need of this paper was used the Matlab function “exmp1” which enables exact calculation of the expression (5) based on Taylor‘s decomposition.

3.3 The numeration substance of probability of a non-functional state of a node

The probability of a non-functional state of a node of an AG, for which the individual input edges are independent, is in fact given by going over all possible combinations of probabilities of the input edges (such combinations that case failure of the node). For 20 input edges we have regularly a million combinations.

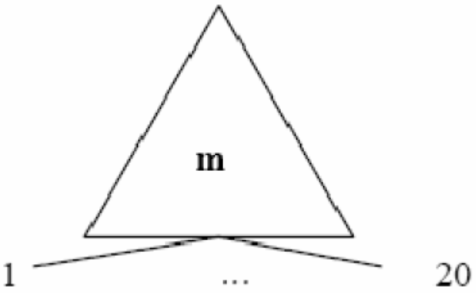


Figure 1. One node of the acyclic graph with 20 edges.

One partial contribution to the probability of a non-functional state of the node in Figure1 has a form:

$$q_1.q_2...q_{i-1}p_i.q_{i+1}...q_{j-1}.p_j.q_{j+1}...q_{20},$$

where a number of occurring probabilities p (here the number equals to 2) can not reach “m”. This fact is in the context of definition of the internal node of AG [1], which is correctly functioning just in the case when at least **m** inferior nodes (either terminal or non-terminal) are correctly functioning.

The probability of a non-functional state of the node is generally given by a sum of a big quantity of very small numbers. These numbers are generally very different!

If the sum will be carried out in the way that the addition runs in the order from the biggest one to the smallest ones, certainly a lost stems from rounding off, more than the addition runs in the order from the smallest ones to the biggest values. And even in this second case there is not possible to determine effectively how much accuracy "has been lost".

Note: In the case of dependence of the input edges (terminal nodes) we cannot express the behaviour of an individual node numerically. There is necessary to work with the whole relevant sub-graph. Combinatorial character for the quantification will stay nevertheless unchanged.

3.4 The error-free sum of different non-negative numbers

The first step to the solution of this problem is to find a method for the "accurate" sum of many non-negative numbers.

The arithmetic unit of a computer (PC) works in a binary scale. A positive real number of today's PC contains 53 valid binary numbers, see Figure 2. A possible order ranges from approximately -1000 to 1000.

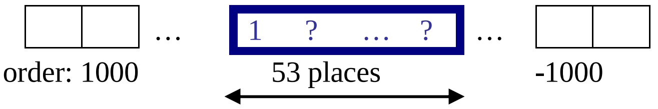


Figure 2. A positive real number in binary scale.

The line indicated as "order" means an order of a binary number.

The algorithm for the "accurate" quantification of sums of many non-negative numbers consists from a few steps:

1. The whole possible machine range of binary positions (bites) is partitioned into segments of 32 positions for orders, according to the following scheme in Figure 3:
2.

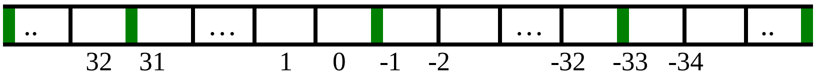


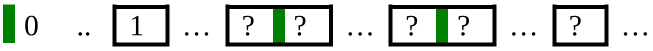
Figure 3. Segments composed from 32 binary positions.

The number of these segments will be approx.:

$$\frac{2000}{32} \cong 63$$

3. Total sum is memorized as one real number, which is composed from 32 bite segments. Each from these segments has additional 21 bites used as transmission.

4. At first a given non-zero number of the sum that must be added is decomposed according to before assigned firm borders (step 1) mostly into three parts containing 32 binary numbers of the number at most, according to the scheme in Figure 4. The individual segments are indexed by numbers 1-63.



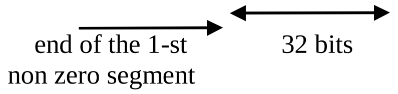


Figure 4. Decomposition of a given non-zero number

5. Then the individual parts of this decomposed number are added to the corresponding members of the sum number, as in Figure 5.

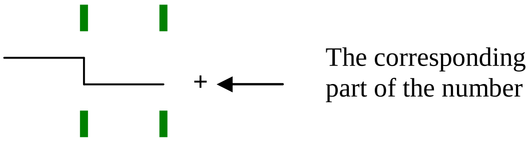


Figure 5. Adding a number to the sum number

6. Always after the processing of 2^{21} numbers (the limit is chosen so that it could not lead to overflowing of the sum number at any circumstances) a modification of the sum number is carried out which is indicated as the “clearance” process. Upwards a transmission is separated (in the following Figure 6 it is identified by a symbol β) which is added to the upper sum.

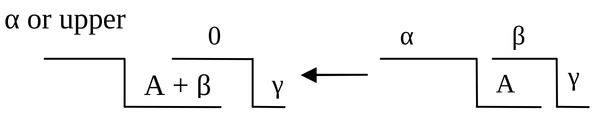


Figure 6. Clearance process.

7. If a final sum is required, at first the clearance process has to be carried out. Then the group of three sums is elaborated, from which the upper is the highest non-zero one (identified by a symbol α in Figure 7). We make a sum of these three sums as usual in a binary scale, when *p* in the following expression is given by an index of the highest non-zero segment:

$$sum = \alpha .2^{p} + \beta .2^{p-32} + \gamma .2^{p-64}$$

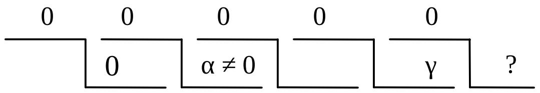


Figure 7. Demonstration of the final summarization.

So numbers in their full machine accuracy (53 binary numbers beginning with 1) are the input for this process of adding. The output is the only number in the same machine accuracy (53 binary numbers beginning with 1). The number is mechanically the nearest number to the real accurate error-free sum which contains in principle up to 2000 binary numbers.

3.5 Permissible context of the usage not leading to the loss of accuracy

The probability of a non-functional state of a repairable component (repairable component with hidden failures) is given by the formula (3), which can be simplified as

$$P(\tau)=(1-P_C).\alpha(\tau)+P_C.\beta(\tau),$$

where P_C is the probability of a non-functional state of an element at the moment of the inspection at the beginning of the interval till the next inspection; α, β are non-negative mechanically accurate and numerically expressed functions.

One contribution to the computation of a non-functional state of a node generally has a form

$$q_1.q_2...(1-q_k)....$$

In both cases occurs (*1*- *q*). It has already been explained that when we use *p* ≡ *1*- *q*, it can come to the catastrophic loss of accuracy. A basic question then comes out: Do we have to modify further the stated patterns for the purpose of removing the subtraction? Fortunately not. In the introduced context of the product *(1- q).α*, where *α* is expressed numerically in a full machine accuracy there is no loss in machine accuracy! Thanks to rounding off the final product to 53 binary numbers, lower orders of the expression *(1-q)*, i.e. a binary numbers on 54th place and other places behind the first valid number, can not practically influence the result.

3.6 Determination of system probability behaviour according to a graph structure

Let all elements appearing in the system are independent. The probability of a non-functional state of a system, assigned by the help of AG, is thus simply gained on the basis of estimation of all nodes upwards. For instance for the AG in Figure 8 the following steps have to be made:

- numerical expression of the probability of a non-functional state of terminal nodes, i.e. elements 8,9,10 and 5,6,7
- numerical expression of the probability of a non-functional state of an internal node 4 which
  is given by the following sum:

$$\begin{aligned}&q_8.q_9.q_{10}+(1-q_8).q_9.q_{10}\\&+q_8.(1-q_9).q_{10}+q_8.q_9.(1-q_{10})\\&+q_8.(1-q_9).(1-q_{10})\\&+(1-q_8).q_9.(1-q_{10})\\&+(1-q_8).(1-q_9).q_{10}\end{aligned}$$

- numerical expression of the probability of a non-functional state of an internal node 3 which
  is given by the only item

$$q_5.q_6.q_7$$

- numerical expression of the probability of a non-functional state of a terminal node 2
- numerical expression of the probability of a non-functional state of the highest SS node 1 which is given:

$$\begin{aligned}&q_2.q_3.q_4+(1-q_2).q_3.q_4\\&+q_2.(1-q_3).q_4+q_2.q_3.(1-q_4)\end{aligned}$$

In the case of AG with dependent elements, where every multiple used node causes dependence, the situation is much more complex. We have to decompose a set of nodes to a disjunctive system of mutually independent subsets. The process has been also implemented to the new algorithm.

Note: Internal numeration of nodes is such that the node with a less number can not be inferior to the node with greater number. Nodes are numbered in the decreasing order of numbers.

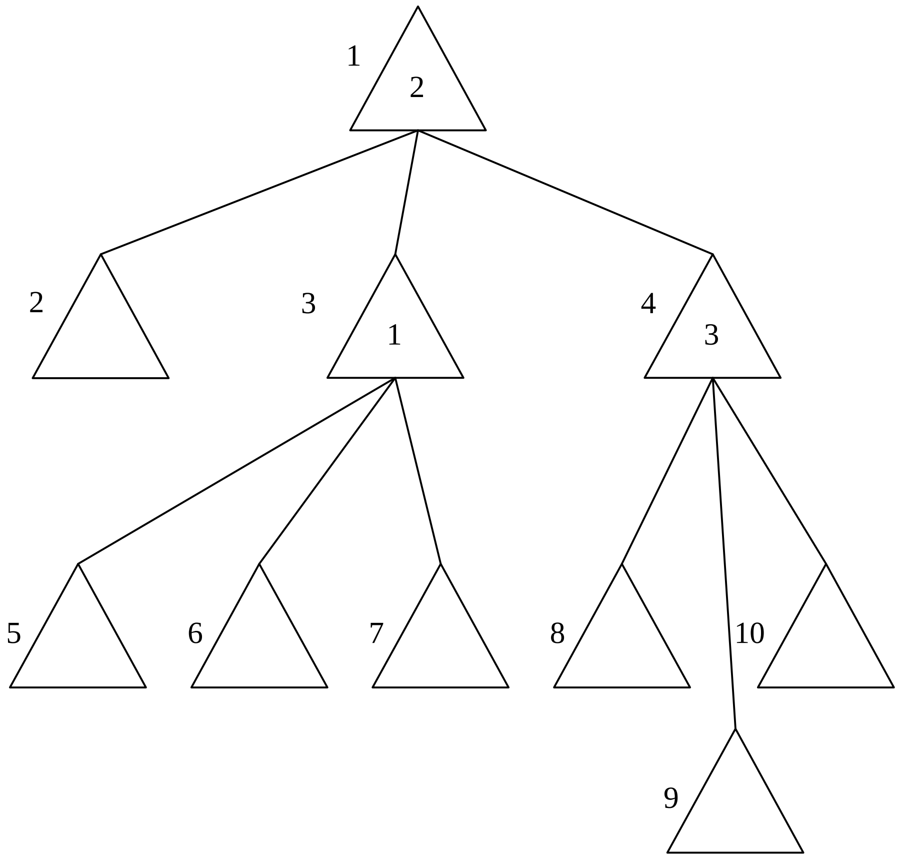


Figure 8: The system assigned by the help of a structure AG

4 EXAMPLE OF COMPUTATIONS WITH HIGHLY RELIABLE SYSTEMS

Let us consider a very reliable electronic system from practice – driving light for a rolling-stock, for example let us consider an electric locomotive. The light of the locomotive is composed from 15 parallel branches of LED diodes. We define the critical failure of the light as follows: the critical failure of the system occurs just in case when tree branches are in failure simultaneously. Other words the critical failure occurs if three out of fifteen branches fail. One branch of LED diodes is repairable component with exponential time to failure with mean 5.10^4. Repair time is also exponentially distributed with mean time to repair 6 hours.

Figure 9 demonstrates the time dependent unavailability coefficient within first 200 hours. Exact computation results show that the unavailability coefficient given by equation (1) was stabilized after 30 hours of operation on a value around 8.10^{-10}.

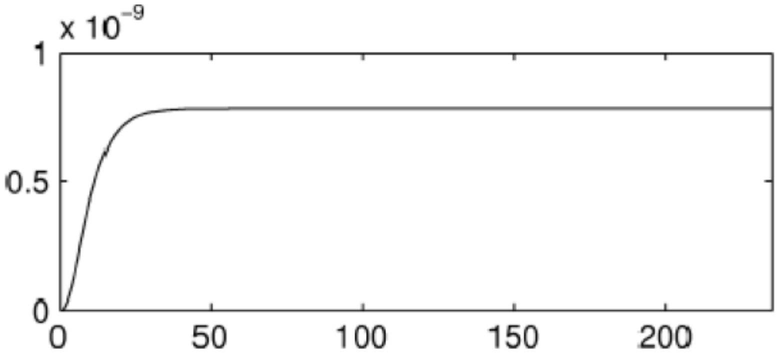


Figure 9: The time dependent unavailability coefficient within first 200 hours.

5 CONCLUSIONS

Maintaining the full machine accuracy requires mainly not to carry out subtraction of near values. All required outputs are therefore necessary to express in the form of the sum of numbers with consistent sign (in our case non-negative).

A problem of a sum of many non-negative numbers can be solved by decomposing a sum into more partial sums which can be carried out without a loss! The process has been numerically realized within a programming environment Matlab.

Numerical expression of probabilities of a non-functional state of one node of an AG has a combinatorial character. We have to go over all combinations of input edges behaviour leading to a non-functional state of the node. The astronomic increase of combinations with the increasing number of elements causes that the program will be usable only up to a certain size of a system. Already at moderate exceeding the critical size of the system it comes to enormous increase of machine time. The computation above run below 1s, on Pentium (R) 4 CPU 3.40GHz, 2.00 GB RAM.

The algorithm enables to carry out exact unavailability analysis of real maintained systems with both preventive and corrective maintenance. The future research will continue with the aim to use the algorithm for maintenance optimization, i.e. to find such a maintenance strategy to minimize the maintenance cost at a prescribed maximal unavailability level.

6 ACKNOWLEDGEMENT

This work is supported by The Ministry of Education, Youth and Sports of the Czech Republic - project CEZ MSM6198910007.

7 REFERENCES

1. Briš R. (2008). Parallel simulation algorithm for maintenance optimization based on directed Acyclic Graph. Reliab Eng Syst Saf 2008;93:852-62.
2. Briš R. and Drábek V. (2007). Mathematical Modeling of both Monitored and Dormant Failures, Proceedings of the 17th Advances in Risk and Reliability Technology Symposium AR2TS, Edited by Lisa Bartlett, Published by Loughborough University, Loughborough, Leicestershire, LE11 3TU, pg. 376-393(2007), ISBN 0 904947 62 9.

3. Bris R. (2008). Exact reliability quantification of highly reliable systems with maintenance. In Safety, Reliability and Risk Analysis: Theory, Methods and Applications – Martorell et al. (eds), 2008 Taylor & Francis Group, pg. 489-496, ISBN 978-0-415-48513-5.