Annals of Nuclear Energy 102 (2017) 91–105

Contents lists available at ScienceDirect

Annals of Nuclear Energy

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

Determination of prime implicants by differential evolution for the dynamic reliability analysis of non-coherent nuclear systems

Francesco Di Maio [a,*], Samuele Baronchelli [a], Matteo Vagnoli [a], Enrico Zio [a,b]

[a] Energy Department, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy
[b] Chair on System Science and Energetic Challenge, Foundation EDF – Electricite de France, Ecole Centrale, Paris, and Supelec, Paris, France

ARTICLE INFO

Article history:
Received 29 July 2016
Received in revised form 12 December 2016
Accepted 16 December 2016

Keywords:
Dynamic reliability
Prime implicants
Non-coherent structure functions
Modified Binary Differential Evolution (MBDE)
Genetic Algorithm (GA)
Binary Differential Evolution (BDE)
Steam Generator (SG)
Nuclear Power Plant (NPP)

ABSTRACT

We present an original computational method for the identification of prime implicants (PIs) in non-coherent structure functions of dynamic systems. This is a relevant problem for dynamic reliability analysis, when dynamic effects render inadequate the traditional methods of minimal cut-set identification. PIs identification is here transformed into an optimization problem, where we look for the minimum combination of implicants that guarantees the best coverage of all the minterms. For testing the method, an artificial case study has been implemented, regarding a system composed by five components that fail at random times with random magnitudes. The system undergoes a failure if during an accidental scenario a safety-relevant monitored signal raises above an upper threshold or decreases below a lower threshold. Truth tables of the two system end-states are used to identify all the minterms. Then, the PIs that best cover all minterms are found by Modified Binary Differential Evolution. Results and performances of the proposed method have been compared with those of a traditional analytical approach known as Quine-McCluskey algorithm and other evolutionary algorithms, such as Genetic Algorithm and Binary Differential Evolution. The capability of the method is confirmed with respect to a dynamic Steam Generator of a Nuclear Power Plant.

1. Introduction

The reliability analysis of systems with significant hardware/-software/human interactions is difficult, because the response of the system under accidental scenarios depends on the time of occurrence and on the magnitude of the events (Zio and Di Maio, 2009; Aldemir et al., 2010). Further, it turns out that the logic of these systems can give rise to non-coherent structure functions, where both failed and working states of the same components can lead the system to failure (Di Maio et al., 2015); for example, if in a system made up of three components J, K, L it fails with components states $(J, \bar{L}, K)$, with the negation sign indicating that the component is failed, whereas it is working when the components states are $(\bar{J}, \bar{L}, K)$, then the system is non-coherent. The traditional Probabilistic Risk Assessment (PRA) modeling tools, e.g. Fault Tree and Event Tree Analysis, have difficulties in including the specific timing and magnitude of the events. On the other hand, so-called dynamic reliability methods can complement the traditional methods to accounts for the interactions among the physical parameters of the processes (temperature, pressure, speed, etc.), the human operators actions and the failures of the components (Aldemir et al., 2010; Siu, 1994; Devooght, 1997; Marseguerra et al., 1998) and to identify the system prime implicants (PIs), i.e., the event product terms that render true the structure function and that cannot be covered by more reduced implicants (Quine, 1952), even if the structure functions are non-coherent.[1] PIs have been introduced as dynamic equivalent of Minimal Cut Sets (MCSs) for conveying the information on the minimum combinations of failures that lead (non-coherent and/or dynamic) the system to failure and that cannot be covered any other implicant (Garrett and Apostolakis, 1999).

Traditionally, non-coherent structure functions have been interpreted as indication of poor system design. However, in Beeson (Beeson, 2002) it is shown that PIs identification can help developing an effective maintenance schedule for non-coherent

[1] For clarity sake, we recall that an implicant is a product of Boolean variables, each one associated with a system component and representing its failed (1) or safe state (0), that leads the system to failure: differently from minterms, in implicants not all the variables have to appear when these (missing) variables cannot affect the system behavior. Implicants, thus, can cover more minterms that differ in only one (or more) variable that does not influence the system failure (as well as cut sets and minterms in traditional PRA).

* Corresponding author.
E-mail address: francesco.dimaio@polimi.it (F. Di Maio).

http://dx.doi.org/10.1016/j.anucene.2016.12.018
0306-4549/

systems. For example, suppose that $\bar{J}, \bar{K}, L$ (components J and K failed and component L working) is a PI that causes a catastrophic system failure. This shows that, if components J, K and L have failed, L should be the last component to be repaired in order to avoid system failure. Furthermore, PIs identification allows taking additional counteracting measures to prevent system failure, for example by forcing failure of component L when component J and K have already failed (Sharvia, 2008).

Fault tree analysis is undoubtedly an useful and efficient tool for minimal cut set identification, but not for PIs identification, since it can only deal with coherent structure functions (Morreale, 1967). The problem of extending the analysis to non-coherent fault trees has, then, been tackled in different ways: the simplification of non-coherent structure functions expressed in canonical forms has been raised by Quine (Quine, 1952) and solved by McCluskey (McCluskey, 1956), allowing a preliminary identification of PIs; the problem has also been tackled by means of graphical methods such as Karnaugh maps (Karnaugh, 1953). However, the actual implementation of these methods becomes very time-consuming when the number of variables involved in the given structure function increases. The computational efficiency has been improved resorting to various Partitioned List algorithms (Morreale, 1970) and fast Binary Decision Diagram (BDD) algorithms Jung et al., 2004: in Worrell et al. Worrell et al. (1981), a modification of a minimal cut sets algorithm known as Simple Prime Implicant Set Algorithm is proposed, although it does not always produce complete PI sets, whereas in Rauzy and Dutuit (Rauzy and Dutuit, 1997) a method is proposed to convert the fault tree of a non-coherent structure function into a BDD for PIs identification, where each of the basic events of the tree is represented as a node with two branches (branch 1 and 0, corresponding to the component failure and working states respectively). This latter approach has been adapted in Bjorkman (Bjorkman, 2013) for PI identification based on Dynamic Flowgraph Methodology (DFM).

The difficulty in developing efficient computational methods for PIs identification lays in the fact that this can be seen as an NP-hard problem of covering a set (the minterms) with elements from given subsets (the PIs) Sen, 1993: each given subset has an associated cost proportional to its dimension and the objective of the problem is to choose the smallest group of subsets whose union contains the whole set with minimal cost, as we shall see in what follows.

In this paper, we develop a new method for identifying all PIs of a non-coherent structure function resorting to the powerful evolutionary algorithm of Differential Evolution (DE) Storn and Price, 1996. The PIs are found by solving by DE a properly defined optimization problem, for determining the exact (not approximated) solution of the Set Covering Problem (SCP) Christofides and Paixão, 1993; Beasley and Chu, 1996: in this way, none of the prime (minimal) failure scenarios (i.e., the PIs) can be neglected by the identification method.

The paper is organized as follows. In Section 2, the artificial case study used to generate the scenarios for the dynamic reliability analysis is presented. In Section 3, the model of a Steam Generator (SG) of a Nuclear Power Plant (NPP) is presented Aubry et al., 2012. In Section 4, PIs identification is formulated as an optimization problem and tackled by resorting to the DE-based approach. In Section 5, the results of the application of the approach to the scenarios of the artificial case and of the SG are presented. Conclusions and remarks are given in Section 6.

2. The artificial case study

For ease of illustration of the method proposed, we build an artificial case study by simulating the accidental scenarios for a system made of 5 components (denoted as A, B, C, D and E), that can fail at random times with random magnitudes, giving rise to different scenarios whose evolutions are represented by 4 monitored signals. Multiple component failures can occur during the system life, set to $T = 7$ [h]. For the simulation, a Monte Carlo sampling procedure for injecting faults of random magnitudes at random times is implemented. In particular, times and magnitudes of faults are obtained by a stratified sampling with respect to the possible accident scenarios (Di Maio et al., 2011). The number of components that fail is sampled from a binomial distribution with parameters $n = 5$ (equal to the number of components) and $p = 0.8$ (so that even rare multiple fault events are included in the set of accident scenarios). The first failure time is sampled from a uniform distribution [0, 1] [h], and the successive failure times are sampled by a stick-breaking strategy from the conditional distributions, uniform from the last sampled time up to 7 [h]. This sampling strategy models a wearing system, with average failure rate increasing in time.The equations deliberately used to simulate the signal evolutions in time during the accidental scenarios are (Table 1):

$$y(t) = 2\alpha_1 a\left[1 + erf\left(\frac{t-\mu}{\sqrt{2}}\right)\right] + 10^{-3\omega} \tag{1}$$

$$y(t) = \alpha_2\left(c^{d^t} - c\right) + 10^{-3\omega} \tag{2}$$

$$y(t) = \alpha_3 bt + 10^{-3\omega} \tag{3}$$

where a, b, c, d, μ, ω, α_1, α_2 and α_3 are randomly sampled from the distributions listed in Table 2. Parameters α_1, α_2 and α_3 represent the magnitudes of the faults of the accidental scenarios. All parameters and variables have arbitrary units.

We take signal 1 as the safety-relevant parameter to be monitored against pre-defined safety thresholds: if it exceeds the upper threshold value of 2.5, the system fails in the "High" end state; if it decreases below the lower threshold value of -1.5, the system end state is "Low" (Baraldi et al., 2013). In Fig. 1, the evolution of the 4 signals for 10 randomly sampled accidental scenarios are shown. Signals measurements are plotted in continuous lines; the upper and lower thresholds are in dotted and dashed lines, respectively.

Fig. 1 shows that under different scenarios, the signals can increase or decrease. This can occur in reality where, for example, if a valve of the coolant injection system of a Nuclear Power Plant

Table 1
Equations used to simulate the signals evolutions in time for each failed component.

Failed component	Signal 1	Signal 2	Signal 3	Signal 4
A	Eq. (1)	Eq. (1)	Eq. (3)	Eq. (1)
B	Eq. (1)	Eq. (2)	Eq. (3)	Eq. (1)
C	Eq. (2)	Eq. (3)	Eq. (1)	Eq. (1)
D	Eq. (2)	Eq. (3)	Eq. (2)	Eq. (1)
E	Eq. (3)	Eq. (3)	Eq. (3)	Eq. (1)

Table 2
Parameters distribution.

Parameter	Distribution	Mean value	Standard deviation
a	Gaussian	0.4	0.017
b	Gaussian	0.4	0.017
c	Gaussian	1.3	0.033
d	Gaussian	1.3	0.017
α_1	Gaussian	1	0.083
α_2	Gaussian	1.05	0.033
α_3	Gaussian	1	0.033
μ	Gaussian	2.45	0.083
ω	Gaussian	0	1