MINIMAL CUT SETS IDENTIFICATION OF NUCLEAR SYSTEMS BY EVOLUTIONARY ALGORITHMS

Francesco Di Maio, Samuele Baronchelli, Enrico Zio

MINIMAL CUT SETS IDENTIFICATION OF NUCLEAR SYSTEMS BY EVOLUTIONARY ALGORITHMS

Francesco Di Maio[1], Samuele Baronchelli[1], Enrico Zio[1,2]

[1] *Energy Department, Politecnico di Milano*

Via Ponzio 34/3, 20133 Milano, Italy

francesco.dimaio@polimi.it

[2] *Chair on System Science and Energetic Challenge*

European Foundation for New Energy – Electricite de France

Ecole Centrale, Paris, and Supelec, Paris, France

ABSTRACT

Fault Trees (FTs) for the Probabilistic Safety Analysis (PSA) of real systems suffer from the combinatorial explosion of failure sets. Then, minimal cut sets (mcs) identification is not a trivial technical issue. In this work, we transform the search of the event sets leading to system failure and the identification of the mcs into an optimization problem. We do so by hierarchically looking for the minimum combination of cut sets that can guarantee the best coverage of all the minterms that make the system fail. A multiple-population, parallel search policy based on a Differential Evolution (DE) algorithm is developed and shown to be efficient for mcs identification, on a case study considering the Airlock System (AS) of CANDU reactor.

1. INTRODUCTION

Fault Tree (FT) is a tool widely used in Probabilistic Safety Assessment (PSA) of Nuclear Power Plants (NPPs) [NUREG, 1983; NASA, 2002; Zio, 2007]. Traditionally, FTs are used for quantifying various probabilistic measures (including probabilities and/or frequencies of sequences, safety margins, importance factors and sensitivity indices) [Kumamoto et al., 1996; Epstein et al., 2005]. The size of the system may challenge the FT analysis, in practical situations: even for the Airlock System (AS) of a CANDU nuclear reactor with only 9 components [Lee et al., 2012], the minimal cut sets (mcs) identification problem gives rise to a FT structure function composed by 2^9=512 minterms (i.e., a product of the literals α representing each component state, 1 failed, 0 safe) , 16867 cut sets Π (i.e., a combination of components failures leading the system into failure state) and cut set chart (i.e., a table with all the minterms as columns and the cut sets as rows) of 8635904 elements. To overcome the problem, research efforts have developed in two directions: one looking for approximations of the probabilistic measures of interest obtained by considering only some selected mcs; another one developing computational methods to easily more efficiently assess the probabilistic measures from the exact mcs. One example of approximation consists in considering only small order mcs (i.e., mcs formed by a small number of elements) [Rauzy, 2001], which in principle capture the main part of the top-event probability. Another truncation process selects only the mcs with probability of occurrence larger than a given threshold. However, mcs truncation can have direct consequences on the safety level of the NPP, because it is not known how many are the mcs neglected (because of big or of small probability) in the estimation of the risk/safety indicators of interest. For this reason, it has been pointed out that mcs exact identification (rather than truncation) is one of the technical issues to be tackled in the development of PSA for risk-informed decision making, e.g. for maintenance, service inspections and safety margins quantification in new NPPs design [Fleming, 2003; Duflot et., al, 2009; Zio et al., 2010].

A first attempt in developing computational methods for limiting the mcs combinatorial explosion of FTs without approximation has been to encode the Boolean formulae derived by the FTs into binary decision diagrams (BDDs) [Akers, 1978]. One of the major advantages of a BDD is that it provides exact values for probabilistic measures and it does not need any kind of truncation or approximations. However, BDD is highly memory consuming and very large models, such as FTs of NPP systems are beyond capability [Rauzy et al., 1997]. Another attempt for identifying mcs is the Dynamic Flowgraph Methodology (DFM), which is a directed graph based approach to model and analyze the behavior of dynamic systems [Garrett et al., 1995]. The main drawback is scalability, that is, the fact that realistic modeling causes a combinatorial explosion as the number of states in the system increases [Bjorkman, 2013]. In order to tackle this challenge, a DFM has been solved by a BDD (based on meta-products or on zero-suppressed BDD) [Bjorkman, 2013]. Also Petri nets suffer from the combinatorial explosion of the number of states, when applied to complex systems [Labeau et al., 2000].

We present a novel approach to tackle this issue of exact mcs identification of coherent and non-coherent FTs based on a Hierarchical Differential Evolution (HDE) algorithm. The ordinary DE algorithm has been demonstrated to be an efficient, effective and robust method for the identification of prime implicants (PIs) in simple non-coherent structure functions [Di Maio et al., 2013]. Here it is applied within a hierarchical scheme to deal with its computational limitations and avoid any approximation in the identification of mcs of complex coherent structure functions. With the proposed scheme, we look for the minimum combination of cut sets that can guarantee the best coverage of all the minterms that make the system fail: during the first step of the iteration process, a multiple-population, parallel search policy is implemented to expedite the convergence of the second step of the exploration algorithm.

The paper is organized as follows. Section 2 is devoted to recalling some base terminology (FT, Boolean Formulae, coherent and non-coherent structure functions, minterms, etc.). In Section 3, the HDE technique for mcs identification is presented. In Section 4, it is applied to the FT of a CANDU Airlock System (AS) and its results are compared with those obtained with a DE algorithm. Conclusions and remarks are given in Section 5.

2. TERMINOLOGY

In this Section, we introduce the terminology used throughout the article with reference to FT analysis. The causal relations that lead to the FT top event can be described by a set of Boolean formulae built over a set of variables (literals) α_1, α_2, α_3,... α_n, and connectives (and, or, not, *k*-out-of-*n*), whose semantics is defined by means of truth tables. By manipulation of the truth tables, the top event can be expressed in terms of primary events (e.g. components failures in our case of interest). The simplest way to express the structure function Φ, which relates the top event to the primary events, is in terms of minimal cut sets (mcs) $\Pi^* \in \Omega$. A mcs is a combination of primary events (cut set Π), which if all, and only all, verified cause the top event to occur. Then, a mcs Π^* is one of the 2^n products of literals (minterms), whose occurrence ensures the failure of the system $\Phi(\Pi^*) = 1$, while no proper subset of Π^* is a cut set [Epstein, 2005]. A structure function Φ is coherent if it can be expressed without any Π^* of complemented literals $\bar{X}$, non-coherent otherwise.

3. CANDU AIR LOCK SYSTEM ANALYSIS

The Airlock System (AS) of a Canada Deuterium Uranium (CANDU) reactor is a safety system required to keep the pressure of the inner side of the reactor vault lower than the outer side in order to avoid the dispersion of contaminants out of the reactor bay, in case of accident. The system consists of a vessel in the containment wall of the reactor vault, with two doors in order to allow the inspection of the vault: one door opens towards the inside of the reactor vault, the other towards the outside; so, at least one airlock door, whose seals are inflated via the air system, must be closed by a latch with sufficient pressure in the seals to fulfill its safety function.

A FT has been developed for analyzing a scenario that involves a Design Basis Accident (DBA) occurred in 2011 in the AS of a CANDU NPP [Lee et al., 2012]. During the accident, the inflation of the seals switched to the back-up air supply tank and the FT top event is the incapability of the AS to maintain the pressure boundary [Lee et al., 2012]. The possible causes for this top event can be: the pressure equalizer valve fails (V1), doors fail to close because latches are not locked (D1) and seals are cracked or cannot be inflated (S1). The pressure equalizer valves are designed to equalize the pressure between the reactor bay and the service side and, therefore, to allow controlled flow between the reactor bay and service side. The pressure equalization can fail due to gear box failure (G1) that may limit the vents from opening and closing, to the presence of leakages in the piping system (P1/P2) or to the failure of the exhaust pipe (E1). The airlock doors must be closed by a latch, otherwise the pressure equalizer valves and seals cannot be called in operation on demand. In addition, the possibility is considered that the back-up tank is already empty (T1) or fails to engage (T2) when the inflation of the seals is switched to the back-up air supply system. The basic failure events that can give rise to the AS failure are listed in Tab. 1.

	Basic Failure Events	**ID Code**
1.	Pressure equalizer valve is failed	V1
2.	Doors fail to close and lock	D1
3.	Seals are cracked	S1
4.	Gearbox fails	G1
5.	The piping system presents minor leakages	P1
6.	The piping system presents major leakages	P2
7.	Exhaust pipe fails open	E1

8.	Back up tank is empty	T1
9.	Back up tank fails to engage	T2

Table 1. *Basic failure events and ID code for the considered DBA in a CANDU AS*

The FT for the DBA here considered is shown in [Lee et al., 2012]. The structure function, whose expression is $\Phi = \left[(G1 \text{ AND } E1) \text{ OR } \left((T1 \text{ AND } (S1 \text{ OR } V1 \text{ OR } P1)) \text{ OR } (V1 \text{ OR } T2 \text{ OR } P2) \right) \text{ OR } D1 \right]$, entails 497 minterms leading to the system failure and 16867 cut sets can be found. The true solution $\bar{x}_{opt}$ comprises 7 mcs Π^* ({D1}, {P2}, {T2}, {V1}, {E1,G1}, {P1,T1}, {S1,T1}) [Lee et al., 2012]. The results of the identification of the mcs of the CANDU AS obtained by the proposed HDE are presented in the following paragraphs.

4. HIERARCHICAL DIFFERENTIAL EVOLUTION FOR MCS IDENTIFICATION

We treat the problem of mcs identification as a set covering problem (SCP) [Beasley et al., 1996]. In the context of mcs identification, the SCP is the problem of covering each one of the minterms by a group of cut sets of minimal cost. We define the cost of a cut set Π as the number of literals α associated with system components included in the cut set (literal cost). Each solution of the SCP, $\hat{\bar{x}}_{opt}$, is represented by a specific combination of independent variables, or, mathematically speaking, by a *R*-dimensional vector $\bar{x} = (x_1, x_2, ..., x_R)$ (hereafter called chromosome within the jargon of the differential evolution (DE) optimization method (Appendix A)) where a value of 1 in the *i*-th vector position $x_i$implies that Π_i is chosen to be in the cover; a value of 0, otherwise [Sen, 1993].

The novelty of the Hierarchical Differential Evolution (HDE), here proposed for mcs identification, builds on in the application of a two-step Differential Evolution (DE) optimization[Wang et al., 2010]. In the proposed hierarchical framework, the first optimization is fed with subsets Γ_i, *i=1, 2, ...,S*, of the whole set Ω, where the *i*-th subset Γ_i is generated by randomly assigning to it *N* cut sets Π of Ω in a way that each cut set belongs to only one subset, i.e., $\Gamma_i \cap \Gamma_j = \varnothing$ for $\text{i} \neq j$, and the union of all the subsets is equal to all the cut sets, i.e. $\bigcup_{i=1}^{n} \Gamma_i = \Omega$.

For each of the subsets Γ_i, *i=1, 2, ...,S,*

1a) we build a cut set chart, using all the minterms as columns and the cut sets Π belonging to Γ_i as rows

2a) we build the cost vector, where to each Π is assigned its literal cost

3a) we perform the DE optimization

4a) we find the best individuals $\{\Pi\}_i$.

The second DE optimization is performed on the new subset $\bigcup_{i=1}^{S}\{\Pi\}_i$ comprising all the cut sets included in the best individuals $\{\Pi\}_i$ found at the end of the first optimization. In detail,

1b) we build a new cut set chart, using all the minterms as columns and the cut sets $\bigcup_{i=1}^{S}\{\Pi\}_i$ as rows

2b) we build a new cost vectorwhere to each $\bigcup_{i=1}^{S}\{\Pi\}_i$ is assigned its literal cost

3b) we perform the DE optimization

4b) we find the mcs Π^* of the system

The pseudo-code is shown below.

```
for $i=1:S$
        sample without replacement $N$ cut sets $\Pi$ from $\Omega$
        populate the $i$-th subset $\Gamma_i$
end
for $i=1:S$
First level DE
     create an initial population of $NP$ potential solutions $\bar{x}$ containing a selection of the cut sets $\Pi$ in the
          $\Gamma_i$ subset
     for $g=1:G$
        select (for each potential solution $\bar{x}_g$) three randomly chosen individuals for reproduction (Eq. 1a)
        create for each target vector $\bar{x}_g$ a noisy vector $\bar{v}_g$ using mutation process
        create a trial vector $\bar{u}_{g+1}$ mixing target and noisy vectors, $\bar{x}_g$ and $\bar{v}_g$, respectively (Eq. 3a)
        compare target vector $\bar{x}_g$ with each related trial $\bar{u}_{g+1}$ and eventually replace (Eq. 4a)
     end
memorize the Π contained in the $i$ best solutions $\{\Pi\}_i$
end
create an initial population of $NP$ potential solutions $\bar{x}$ composed by $\{\Pi\}_i$, $i=1,2,\dots,S$
Second level DE
     for $g=1:G$
        select (for each potential solution $\bar{x}_g$) three randomly chosen individuals for reproduction (Eq. 1a)
        create for each target vector $\bar{x}_g$ a noisy vector $\bar{v}_g$ using mutation process
        create a trial vector $\bar{u}_{g+1}$ mixing target and noisy vectors, $\bar{x}_g$ and $\bar{v}_g$, respectively (Eq. 3a)
        compare target vector $\bar{x}_g$ with each related trial $\bar{u}_{g+1}$ and eventually replace (Eq. 4a)
     end
```

```
memorize the best solution found $\hat{\bar{x}}_{opt}$
```

Three performance indicators are used to judge the goodness of the results. In the evaluation, the optimizations are repeated a number of times (20 in our case), to account for the inherent stochasticity of the search algorithm. The three performance indicators are:

- Cpu: cpu time (expressed in seconds) necessary to converge to the solution $\hat{\bar{x}}_{opt}$.
- Success rate (Sr): percentage of trials for which the true optimum $\bar{x}_{opt}$ is found.
- Accuracy (λ): the larger λ, the larger the accuracy of the solution [Tvrdik, 2006] as:

$$\text{if } \bar{x}_{opt} \neq 0 \qquad \lambda = \begin{cases} 0 & \text{if } \dfrac{\left|\hat{\bar{x}}_{opt} - \bar{x}_{opt}\right|}{\left|\bar{x}_{opt}\right|} \geq 1 \\ 11 & \text{if } \dfrac{\left|\hat{\bar{x}}_{opt} - \bar{x}_{opt}\right|}{\left|\bar{x}_{opt}\right|} < 10^{-11} \\ -\log_{10}\left(\dfrac{\left|\hat{\bar{x}}_{opt} - \bar{x}_{opt}\right|}{\left|\bar{x}_{opt}\right|}\right) & \text{otherwise} \end{cases} \tag{1}$$

$$\text{if } \bar{x}_{opt} = 0 \qquad \lambda = \begin{cases} 0 & \text{if } \left|\hat{\bar{x}}_{opt}\right| \geq 1 \\ 11 & \text{if } \left|\hat{\bar{x}}_{opt}\right| < 10^{-11} \\ -\log_{10}\left(\left|\hat{\bar{x}}_{opt}\right|\right) & \text{otherwise} \end{cases}$$

4.1 DE Results

We apply the DE approach proposed in [Di Maio et al., 2013]. A “One complement” fitness function [Shackleford et al., 2001] is embedded into the evolutionary algorithm: the cost of the trial solution $\hat{\bar{x}}_{opt}$ is mapped into a binary fitness function made up by two parts where the most important digits are determined as the complement to one of the uncovered faulty minterms, whereas the least important digits are determined as the complement to one of the sum of the costs of the cut sets included in the trial solution. In this way, we obtain that a complete subset of cut sets that covers all faulty minterms has surely a larger fitness than any other incomplete subset. For the ease of clarity, with respect to the AS of the CANDU, since the columns of its cut set chart are 497 (that is, equal to the number of minterms), 9 bits code the maximum number of uncovered columns, whereas the sum of the cost of all the 16867 cut sets is equal to 103298 so that 17 bits code the cost part of the trial solution. In Fig. 1 the calculation procedure of the "One Complement" fitness function is shown for the best solution: the uncovered columns are equal to zero, while the total cost of the best solution is equal to 10 (4 cut sets contain only one basic event and 3 contain 2 basic events); the complement to one of 0 on 9 bits is equal to 511, and the complement to one of 10 on 17 bits is equal to 131061; joining together this two parts of the fitness function gives a fitness value for $\bar{x}_{opt}$ equal to 67108853.

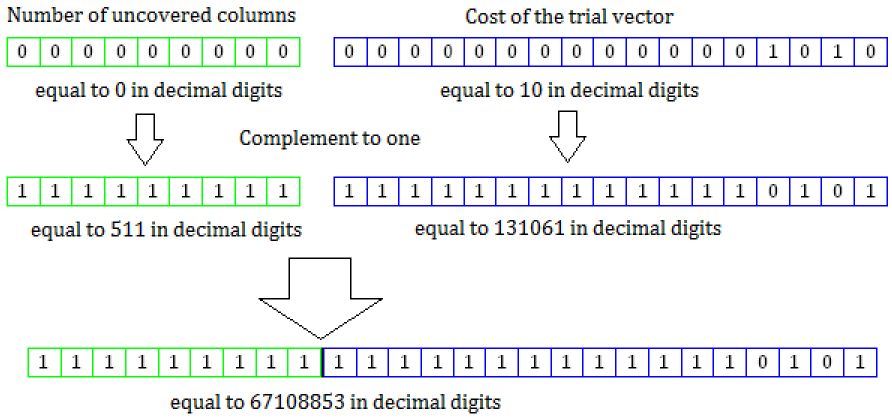

Fig. 1. *Procedure for the calculation of the fitness function for the best solution of the CANDU AS*

In this application, parameters *F* and *b* (Eq. 1a) and *CR* (Eq. 3a) are set equal to the values reported in Tab. 2.

Parameters	***F***	0.1
	CR	0.2
	b	9

Table 2. *Values of the parameters F, CR and b used in the DE*

The analysis is performed for population sizes *NP*=30, 100, 300 and 500. The only stopping criterion is the generation number *G* set equal to *MAXGEN* =5000. Performance indicators for the DE optimizations are shown in Tab. 4, 5, 6 and 7, for *NP*=30, 100, 300 and 500, respectively.

NP	30
Cpu [s]	2046.63
Sr	0 %
λ	4.50

Tab. 4. *Performance indicators for the DE performed with NP=30*

NP	100
Cpu [s]	4449.50
Sr	0 %
λ	6.24

Tab. 5. *Performance indicators for the DE performed with NP=100*

NP	300
Cpu [s]	10092.59
Sr	5 %
λ	7.57

Tab. 6. *Performance indicators for the DE performed with NP=300*

NP	500
Cpu [s]	17503.51
Sr	75 %
λ	10.20

Tab. 7. *Performance indicators for the DE performed with NP=500*

It is seen that, in this real case with a large number of minterms and cut sets, the *NP* value is critical for finding the true mcs list: for population sizes *NP*=30 and 100, the DE cannot succeed in identifying the correct mcs (Sr=0), whereas with *NP*=300 the Sr increases only to 5%. Even the success rate for *NP*=500 (Sr=75%) cannot be acceptable for industrial applications or regulatory purposes.

Tab. 8 reports the results obtained with *NP*=700 and limiting *G* to *MAXGEN* =2000 which are sufficient to achieve convergence and allow saving computational time, i.e. better cpu performance, whilst expediting the convergence towards the optimal mcs $\hat{x}_{opt}$ because of a larger population. In fact, when *NP*=700, the Sr and λ indicators outperform those achieved with smaller population sizes.

NP	700
Cpu [s]	9353.48
Sr	100 %
λ	11

Tab. 8. *Performance indicators for the DE performed with NP=700*

It is important to notice that, with this setting:

1) Sr turns out to be equal to 1, which means that the algorithm always finds the true optimum solution $\hat{\bar{x}}_{opt}$;
2) cpu indicator shows that reducing the value of *MAXGEN* allows for an effective practical application.

In Fig. 2, the evolution of the difference Δ between the fitness values of $\bar{x}_{opt}$ and $\hat{\bar{x}}_{opt}$ during 20 different trials of the iterative search for the optimal solution $\hat{\bar{x}}_{opt}$ is shown on a semilogarithmic plot for *NP*=30, 100, 300, 500 and 700: the mean values of the differences Δ at each generation are plotted in continuous line, with error bars of the minimum and maximum fitness values at each generation.

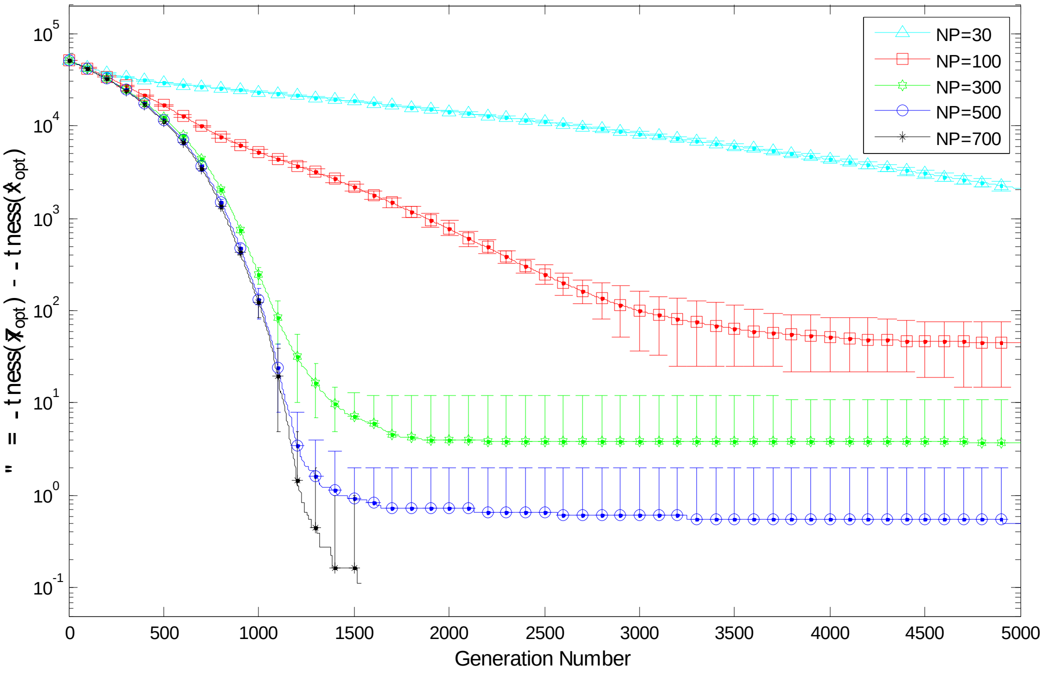

Fig. 2. *Fitness function convergence using DE with different NP values*

The convergence of the algorithm improves in value and number of generations for larger population sizes because of a better exploration of the search space, as confirmed by the Sr values of the indicators for the different *NP* values in Tables 4 to 8 and in Figs. 3 and 4.In particular, in Fig. 3, the boxplots of the $\hat{\bar{x}}_{opt}$ fitness values obtained by the ordinary DE algorithm (with population of 30, 100, 300, 500 and 700 chromosomes) after MAXGEN generations are plotted. For the sake of clarity, in Fig. 4 the plotted boxplots are the zoom of Fig. 2 with respect to population sizes of *NP*=100, 300, 500 and 700.

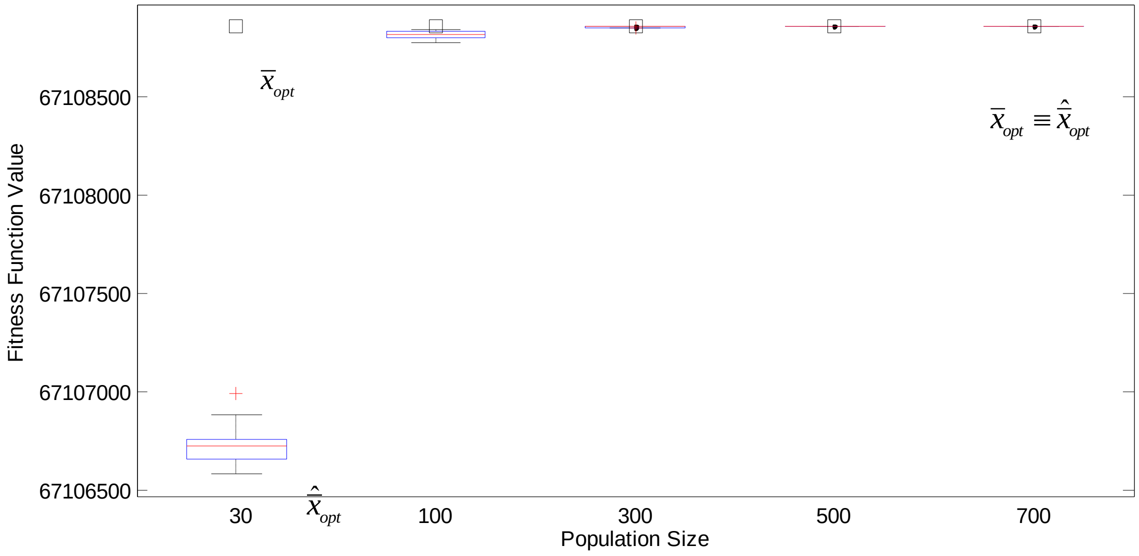

Fig. 3. *Boxplot of the fitness values obtained with the DE-based algorithm with NP=30, 100, 300, 500 and 700*

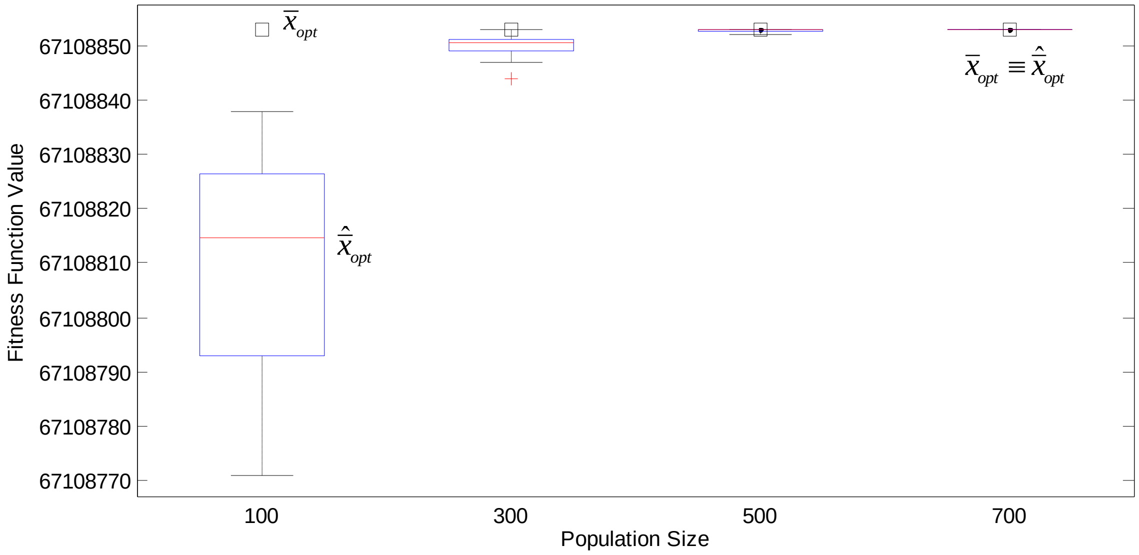

Fig. 4. *Boxplot of the fitness values obtained with the DE-based algorithm with NP=100, 300, 500 and 700*

It can be noticed that increasing the population size *NP* moves the mean fitness value of the population towards the fitness value of the true optimum solution equal for this case study to 67108853. Moreover, the increase of the number of individuals in the population gives rise to distributions that are shrinked on the $\bar{x}_{opt}$ fitness value, which makes the result more reliable.

In conclusion, *NP* reveals to be a critical parameter for the efficiency of the search of the mcs by the DE algorithm: the broader the search space (i.e., the number of cut sets) the larger *NP* is needed to explore it. On the other hand, Tabs. 4 to 8 show that the cpu indicator increases less than linearly with the population size, whereas it is linearly dependent on the number of generations. Therefore, a compromise must be sought to improve the cpu by using a larger population for a shorter number of generations.

4.2. HDE Results

In order to apply the HDE technique to the CANDU FT, we have to partitioned Ω into S subsets Γ_i, where S is set to 10 and N=1687 for eachsubset. The cut set chart and cost vector are built for each subset as shown in Section 3.1. Then, the DE algorithm embedded into the first optimization stage is run with a population size *NP* equal to 500 and the maximum generation number *MAXGEN* equal to 700. In Tab. 9 the mean cpu time required for performing the optimization on a single *i*-th subset is shown.

NP	500
MAXGEN	700
Cpu [s]	607.20

Tab. 9. *Cpu time for the first step of the HDE optimization*

At this first optimization stage, it is not possible to apply the Sr and λindicators because the true solution $\bar{x}_{opt}$ does not belong to any of the subsets. With respect to the computational demand of the first stage of the HDE optimization, it is worth considering that on ordinary computers (e.g., Intel® Core™ i5.2500 CPU @3.30GHz) it is possible to treat at least 5 subsets at the same time. Thus, the total time approximately required for the first step is 1214.40 s.

The number of cut sets found by the first step of the optimization is 100, among which the ordinary DE embedded in the second step of the optimization procedure has to search for the optimal mcs set $\bar{x}_{opt}$. As in the first step, the cut set chart and the cost vector associated to the new cut sets are defined, and a DE search is launched with *NP* and *MAXGEN* equal to 500 and 200, respectively (Tab. 10). The mcs found by the HDE are the same as those reported in Tab. 3, proving that the HDE is capable of identifying the mcs. The cpu time required for the second step optimization is equal to 19.76s.

NP	500
MAXGEN	200
Cpu [s]	19.76

Tab. 10. *Cpu time required by the second step of the HDE optimization*

The total time required by the HDE optimization is equal to 1233s, which is significantly reduced compared with the previous DE (see Tabs. 4 to 8).

The faster convergence obtained by the HDE compared with previous DE with *NP*=700 is shown in Fig. 5, where the evolution of the difference Δ between the fitness values of $\bar{x}_{opt}$ and $\hat{\bar{x}}_{opt}$ is shown on a semilogarithmic plot. For DE, parameters NP and MAXGEN were set equal to 700 and 2000 respectively, while for HDE they are equal to the values of Tabs. 9 and 10.

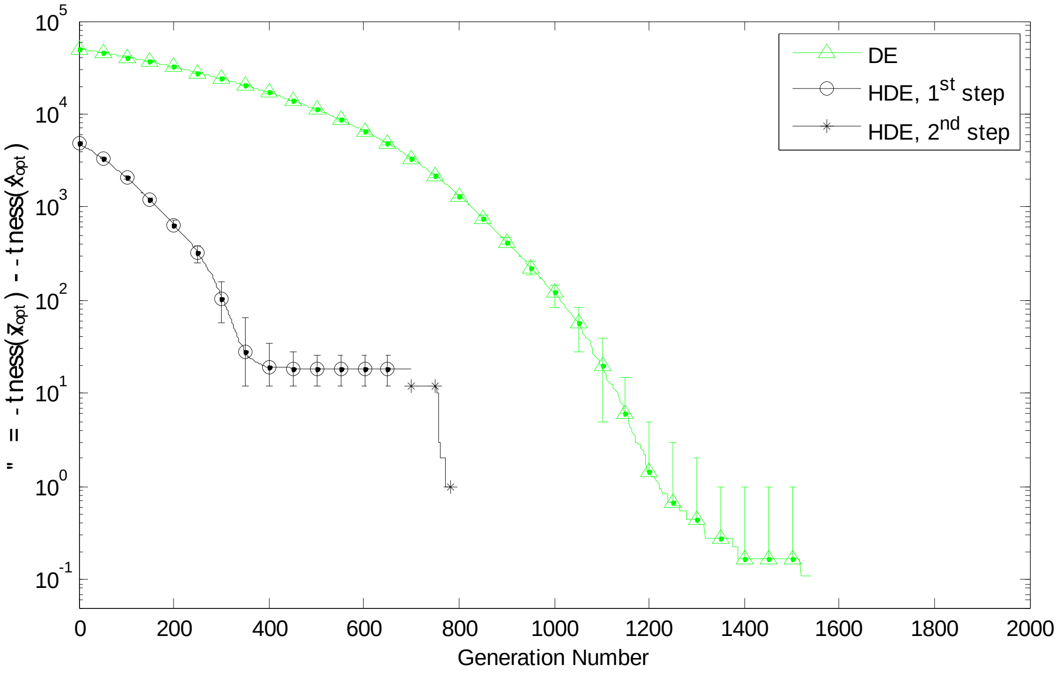

Fig. 5. *Fitness function convergence using DE and HDE*

HDE shows superior performance. In fact, in the first step of the optimization it achieves better results than DE by resorting to a larger population for exploring a reduced search space, whereas in the second step of the optimization it explores an even more reduced search space made up of best individuals, reaching Δ=0 in only 800 generations.

5. CONCLUSIONS

The exact identification of the mcs of FTs is an important task in PSA. It becomes non-trivial for systems that are composed by large numbers of components. In this paper, we have addressed this issue by proposing a novel HDE algorithm. This amounts to transferring the mcs identification into a hierarchical optimization problem: during the first step, a multiple population parallel DE search policy is used to expedite the convergence of a second step of DE exploration. The proposed method has been applied for the analysis an airlock system of a CANDU reactor. The superior HDE results becomes evident as the number of basic events in the FT grows.

APPENDIX A

Differential Evolution (DE)

For solving the so defined SCP we resort to Differential Evolution (DE) [Wang et al., 2010], which belongs to the class of Evolutionary Algorithms (EAs) [Storn et al., 1997][Holland, 1975].

DE entails three phases called mutation, crossover and selection. In the first phase, at the *g+1*-th generation, for each gene $x_r$in the chromosome vector $\overline{x}_g = (x_1, x_2, ..., x_R)_g$ of the population of *NP* different chromosomes at the g-th generation, a probability estimation vector $P(\overline{x}) = \left[P(x_1), P(x_2), \ldots, P(x_R)\right]$ is created by Eq. (1).

$$P(x_r) = \frac{1}{1 + e^{-\frac{2b \square \left[x_{r(l)} + F \square \left(x_{r(k)} - x_{r(m)} \right) - 0.5 \right]}{1+2F}}} \qquad (1a)$$

where the weighting factor $F \in [0, 2]$ is a user-defined parameter, kept constant during the optimization and $x_{r(l)}$, $x_{r(k)}$ and $x_{r(m)}$ are the r-th genes of the three randomly chosen individuals, with $l, k, m \in \{1, 2, ..., NP\}$.

According to the probability estimation vector, the corresponding genes of the noisy vector $\overline{v}_{g+1}$ of the current target individual $\overline{x}_g$ are generated:

$$v_r = \begin{cases} 1 & \text{if } rand \le P(x_r) \\ 0 & \text{otherwise} \end{cases} \qquad (2a)$$

The genes of the trial individual $\overline{u}_{g+1}$ can be obtained by the crossover operator through Eq. (3):

$$u_r = \begin{cases} v_r & \text{if } rand \le CR \text{ or } r = irand(R) \\ x_r & \text{otherwise} \end{cases} \qquad (3a)$$

Therefore, at least one bit of the trial individual is inherited from the mutant individual so that DE is able to avoid duplication individuals and effectively search within the neighborhood; this contributes to maintain the diversity inside the perturbed population, shuffling old and new information. This increases the probability of maintaining some good property from the target vector and avoids drastic changes during the generation of new solution.

During the selection process, the population is modified by substitution. Referring to minimization, if the fitness, i.e., the cost, of $\bar{u}_{g+1}$ is less than the fitness of $\bar{x}_g$, the first will be a member of the next generation g*+1*, replacing the target vector, and the trial vector is discarded

$$\bar{x}_{g+1} = \begin{cases} \bar{u}_{g+1} & if\ fitness(\bar{u}_{g+1}) < fitness(\bar{x}_g) \\ \bar{x}_g & otherwise \end{cases} \tag{4a}$$

The selection criterion in DE is greedy and for sure the following generation is better or at least equal to the previous generation.

References

[Akers, 1978] Akers B., “Binary decision diagrams”, IEEE Transactions on Computers, 276, 509-516, 1978.

[Beasley et al., 1996] Beasley J.E., Chu P.C., “*A genetic algorithm for the set covering problem*”, European Journal of Operational Research, vol.94, 392-404, 1996.

[Bjorkman, 2013] Bjorkman K., “*Solving dynamic flowgraph methodology models using binary decision diagrams*”, Reliability Engineering and System Safety, 111, 206-216, 2013

[Di Maio et al., 2013] Di Maio F., Baronchelli S., Zio E., “*Prime Implicants Determination by Differential Evolution for Dynamic Reliability Analysis of Non-Coherent Systems*”, under review, Reliability Engineering and System Safety.

[Duflot et al., 2009] Duflot N., Bérenguer C., Dieulle L., Vasseur D., “*A min cut-set-wise truncation procedure for importance measures in probabilistic safety assessment*”, Reliability Engineering and System Safety, 94, 1827-1837, 2009.

[Epstein et al., 2005] Epstein S., Rauzy A., “*Can we trust PRA?*”, Reliability Engineering and System Safety, 88, 195-205, 2005.

[Fleming, 2003] Fleming K.N., “*Issues and recommendations for advancement of PRA technology in risk-informed decision making*”, Technical Report NUREG/CR-6813, U.S. Regulatory Commission, 2003.

[Garrett et al., 1995] Garrett C., Guarro S., Apostolakis G., “*The dynamic flowgraph methodology for assessing the dependability of embedded software systems”*, IEEE Transactions on Systems, Man and Cybernetics, 25, 824-840, 1995.

[Holland, 1975] Holland J.H., “*Adaptation in Natural and Artificial Systems*”, University of Michigan Press, Ann Arbor, 1975.

[Kumamoto et al., 1996] Kumamoto H., Henley E.J., “*Probabilistic risk assessment and management for engineers and scientists*”, New York, IEEE Press, 1996.

[Labeau et al., 2000] Labeau P.E., Smidts C., Swaminathan S., “*Dynamic reliability: towards an integrated platform for probabilistic risk assessment”,* Reliability Engineering and System Safety, 68, 219-254, 2000.

[Lee et al., 2012] Lee A., Lu L., “*Petri Net Modeling for Probabilistic Safety Assessment and its Application in the Air Lock System of a CANDU Nuclear Power Plant*”, Procedia Engineering, 2012 International Symposium on Safety Science and Technology, Vol. 25, 11-20, 2012

[Marseguerra et al., 2004] Marseguerra M., Zio E., “*Monte Carlo estimation of the differential importance measure: application to the protection system of a nuclear reactor*”, Reliability Engineering and System Safety, V. 86, 11-24, 2004

[NASA, 2002] Fault Tree Handbook with Aerospace Applications, NASA, 2002.

[NUREG, 1983] PRA Procedures Guide, Vols 1&2, NUREG/CR-2300, 1983.

[Petersen, 1981] Petersen J.L., “*Petri net theory and the modeling of systems*”, Englewood Cliffs, NJ: Prentice-Hall, 1981.

[Rauzy et al., 1997] Rauzy A., Dutuit Y., “*Exact and truncated computations of prime implicants of coherent and non-coherent fault tree*”, Reliability Engineering and System Safety, 58, 127-144, 1997.

[Rauzy, 2001] Rauzy A., “*Mathematical Foundations of Minimal Cutsets*”, IEEE Transactions on Reliability, Vol. 50, No. 4, 2001.

[Sen, 1993] Sen S., “Minimal cost set covering using probabilistic methods”, Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing: states of the art and practice, 157-164, 1993.

[Shackleford et al., 2001] Shackleford B., Snider G., Carter R.J., Okushi E., Yasuda M., Seo K., Yasuura H., “*A High-Performance, Pipelined, FPGA-Based Genetic Algorithm Machine*”, Genetic Programming and Evolvable Machines, Volume 2, Number 1, 33-60, 2001.

[Schreiber et al., 2009] Schreiber R., Theriault K., *Pressurized Water Reactors (PWRs)* and *Boiling Water Reactors (BWRs)* in *Nuclear Engineering Handbook,* edited by K.D. Kok, CRC Press, 2009

[Sen, 1993] Sen S., “Minimal cost set covering using probabilistic methods”, Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing: states of the art and practice, 157-164, 1993.

[Sharvia et al., 2008] Sharvia S., Papadopoulos, “*Non-coherent Modelling in Compositional Fault Tree Analysis*”, Proceedings of the 17th World Congress, The International Federation of Automatic Control, Seoul, Korea, July 6-11, 2008

[Storn et al., 1996] Storn, R.; Price, K. (1997). "*Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces*". *Journal of Global Optimization* 11: 341–359, 1996.

[Tvrdìk, 2006] Tvrdìk J., “*Competitive differential evolution*”, in MENDEL 2006, 12th International Conference on Soft Computing, 7-12, 2006.

[Wang et al., 2010] Wang L., Fu X., Menhas M.I., “*A Modified Binary Differential Evolution Algorithm*”, Life Modelling and Intelligent Computing, Lecture Notes in Computer Science, Volume 6329/2010, 2010.

[Wash-1400, 1976] Wash-1400 (NUREG 75/014), “*Reactor safety study: an assessment of accident risks in US commercial nuclear power plant*”, Appendix 2: Fault Trees, 1976.

[Zio, 2007] Zio E., “*An introduction to the basics of Reliability and Risk Analysis*”, World Scientific Publishing, 2007.

[Zio et al., 2010] Zio E., Di Maio F., Tong J., “*Safety Margins Confidence Estimation for a Passive Residual Heat Removal System”*, Reliability Engineering and System Safety, 95, 828-836, 2010.