Turati P., Pedroni N., & Zio E 2017. An adaptive simulation framework for the exploration of extreme and unexpected events in dynamic engineered systems. *Risk Analysis*, 37(1), 147-159.

An adaptive simulation framework for the exploration of extreme and unexpected events in dynamic engineered systems

Pietro Turati[1], Nicola Pedroni[1], Enrico Zio[1,2,*]

[1] Chaire Systems Science and the Energy Challenge, Fondation Electricité de France (EDF), Laboratoire Genie Industriel, CentraleSupélec, Université Paris-Saclay, Grande voie des Vignes, 92290 Chatenay-Malabry, France

[2] Energy Department, Politecnico di Milano, Via La Masa 34, Milano, 20156, Italy.

[*] Corresponding author: enrico.zio@polimi.it, enrico.zio@centralesupelec.fr

Abstract

The end states reached by an engineered system during an accident scenario depend not only on the sequences of the events composing the scenario, but also on their timing and magnitudes. Including these additional features within an overarching framework can render the analysis infeasible in practical cases, due to the high dimension of the system state space and the computational effort correspondingly needed to explore the possible system evolutions in search of the interesting (and very rare) ones of failure. To tackle this hurdle, in this paper we introduce a framework for efficiently probing the space of event sequences of a dynamic system by means of a guided Monte Carlo simulation. Such framework is semi-automatic and allows embedding the analyst prior knowledge about the system and his/her objectives of analysis. Specifically, the framework allows adaptively and intelligently allocating the simulation efforts preferably on those sequences leading to outcomes of interest for the objectives of the analysis, e.g., typically those that are more safety-critical (and/or rare). The emerging diversification in the filling of the state space by the preference-guided exploration allows also the retrieval of critical system features, which can be useful to analysts and designers, for taking appropriate means of prevention and mitigation of dangerous and/or unexpected consequences. A dynamic system for gas transmission is considered as case study to demonstrate the application of the method.

Keywords: dynamic event tree; unexpected accident scenarios; random exploration; Markov Chain Monte Carlo (MCMC); Integrated Deterministic Probabilistic Safety Assessment (IDPSA).

1 Introduction

The dynamic response of an engineered system under different conditions can be studied, in general, by means of mathematical models implemented in corresponding computer codes for numerical simulations. In particular, in the analysis of safety-critical systems, such as nuclear power plants, oil and gas facilities, electrical grids, etc. model simulations are used to identify extreme configurations and avoid that they remain unexplored and unknown until their (possibly catastrophic) occurrence (Gao, Liu, & Dougal, 2002; Hansen, Jauch, Sarensen, Iov, & Blaabjerg, 2004; Smidts & Devooght, 1992). The outputs of the simulations guide the analysis of the system evolutions and the identification of those event sequences (i.e., scenarios) that can lead the system into extreme conditions (Bier, Haimes, Lambert, Matalas, & Zimmerman, 1999; Pate-Cornell, 2002; Paté-Cornell, 2012). In this context, the combination of Event Trees (ETs) (representing the logic of the system) and mathematical models of the system dynamics (describing the dynamics of the physical phenomena involved) has been largely advocated as the way for determining the End States (ESs) that can be reached by the system and deriving the corresponding causality relations among the events (Aldemir, 2013; J. H. Li et al., 2011; Siu, 1994).

In this line of thought, works on Dynamic Event Trees (DETs) (Čepin & Mavko, 2002; Cojazzi, 1996; Hakobyan et al., 2008; Hsueh & Mosleh, 1996; Kloos & Peschke, 2006; Labeau et al., 2000), have highlighted that the end states reached by a system as a result of an accident scenario do not depend only on the order of occurrence of the events in the sequence, but also on the exact time at which these events occur and on their magnitude (Aldemir, 2013; Devooght & Smidts, 1992; F. Di Maio et al., 2015a; Garrett & Apostolakis, 1999; J. H. Li et al., 2011). However, the introduction of the time and magnitude dimensions into the analysis makes the size of the system state space theoretically infinite and, thus, impossible to be explored completely. Also (and in any case), the computational time needed for running a single simulation of the system evolution can be significant: consider, for example, the computer code RELAP used to simulate the thermo-hydraulic behavior of nuclear systems, which can take hours or days for a single run in specific conditions (Fong et al., 2009; Perez et al., 2011; RELAP5-3D, 2005). To address this issue, the majority of the methods available in the literature exploits the discretization of the time dimension and the pruning of ET branches that have low probability, for the purpose of reducing the number of possible event sequences to be explored; however, these techniques may miss “rare” sequences leading to extreme safety-critical outcomes, as pointed out in (Hakobyan et al., 2008; Rutt et al., 2006), where the authors consider the possibility of biasing the exploration toward critical events.

In view of these challenges, efficient methods for Integrated Deterministic Probabilistic Safety Assessment (IDPSA) are currently being developed to take into account time-dependences in the

evolution of the dynamic system and to probe the corresponding event sequence space for identifying unknown unreliability, unexpected scenarios and critical configurations (Zio, 2014). Along this line of research, this paper contributes an efficient framework for exploring the state space of a dynamic system with the purpose of discovering event sequences leading to unexpected outcomes. In this context, (Hu et al., 2004; Turati et al., 2015) propose methods that focus the exploration efforts, i.e. the simulations, on those scenarios having more uncertain outcomes (i.e., a higher number of end states). For this purpose, they exploit a function based on negative entropy for assessing the uncertainty in the outcomes and a Bayesian scheme for updating the knowledge gathered by the simulations (T. M. Cover & Thomas, 2006; Hu, 2005; Mackay, 1992). As a result, scenarios that can reach a larger number of ESs are explored more frequently and thoroughly.

In this paper, new driving functions are proposed to allow embedding the analyst knowledge and preferences into the exploration, e.g., the interest on specific scenarios or ESs. The new driving functions are implemented within a novel, adaptive, semi-automatic exploration framework for efficiently probing the system state space and retrieving the corresponding information of interest. Demonstration is given with regards to a simple, but representative, dynamic system made by a gas transmission pipe actively controlled by a valve and connected in series to two other pipes in parallel. All the system components are subject to stochastic failures, described by assigned probability distributions. The results of the method are compared to those of a crude Monte Carlo Sampling and an entropy-based search scheme of the system state space (Turati et al., 2015).

The rest of the paper is organized as follows. In Section 2, a general description of the problem under analysis is given. In Section 3, the Semi-Automatic Adaptive Exploration Framework is presented. In Section 4, the case study and the associated results are reported. Finally, in Section 5, some conclusions are drawn and prospective research challenges are suggested.

2 Problem Definition

Within the framework of interest of the present paper, a scenario defines a specific sequence (i.e., order) of events in the life evolution of the dynamic system, which may involve a particular group of components, safety functions or actions (e.g., mechanical failures, activation of safety systems and human decisions). In this context, Dynamic Event Tree (DET) is used as the logical modeling technique to derive, by means of simulation, the scenarios that can arise in the life evolution of a dynamic system as the result of a sequence of successes and failures of different components and functions (J. H. Li et al., 2011; Mercurio et al., 2009). Many software for DET analyses, such as DYLAM (Cojazzi, 1996), ADS (Hsueh & Mosleh, 1996) and MCDET (Kloos & Peschke, 2006), are available and, in principle, *all* the possible (accident) *scenarios* could be extracted, especially by

recurring to massive parallel computing (Catalyurek et al., 2010) or to backtracking techniques (J. H. Li et al., 2011). However, *not all* the possible different *time sequences* within a given scenario can be explored, by reason of the extremely high computational cost needed for simulating all of them. This is relevant since, as shown for example in (F. Di Maio et al., 2015a; Francesco Di Maio et al., 2015), different time sequences (even within the same scenario) may lead to different outcomes, hereafter also called End States (ESs). This justifies the research interests and efforts in taking into account the time dimension of the problem. Typically, an end state is a categorical variable representing in a synthetic way the configuration of the system, e.g., the degradation level of the components (i.e., low, medium and high degradation) and the state of the system (safe, warning, failure). For the sake of clarity, let us consider a component, which starts deteriorating once an initiating event (damage) occurs. Then, the state of degradation of the component, i.e., its End State ES, at a given mission time T_{Miss} depends on the time T_{init} of the initiating damage. An illustrative example is reported in Fig. 1.

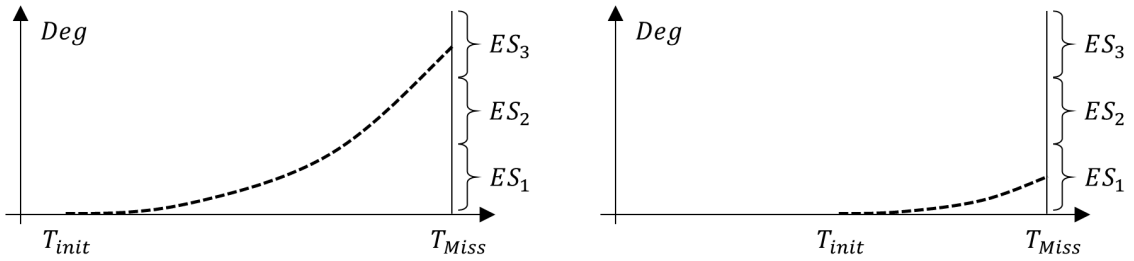

Fig. 1 Illustrative example of the effects that different damage initiating times T_{int} have on the final degradation state of a fictitious component.

However, depending on the number and typology of the events involved in a scenario, the impact of the event times on the system model simulations can be different. In this light, some scenarios may always lead to similar simulations generating the same outcome; on the contrary, other sequences can lead to a larger variability in the outcomes. For this reason, during the exploration of the possible system evolutions, in some cases, it may be in the analyst interest to focus the simulation efforts on those scenarios that show higher variability in the ESs (i.e., that have more uncertain outcomes). To this aim, a novel approach, which takes its root in (Turati et al., 2015), is here proposed. The general procedure consists of two steps: (i) selection of a scenario which is "worth" to be explored according to a predefined "driving criterion"; (ii) simulation of a possible system evolution, conditioned to the selected scenario.

In other situations, instead, the analyst may be already aware of some possible evolutions and he/she may be interested in precisely identifying those time sequences belonging to a given scenario S_j that can lead to a particular ES_i, with the objective of retrieving information and

features of the scenario useful to prevent them and prepare for protection and mitigation of their consequences (F. Di Maio et al., 2015a; Mercurio et al., 2009).

Since the contribution of the paper mainly lies in the efficient and intelligent exploration of the system state space, with no interest in the estimation of the probabilities of the events of interest, in all the situations considered (see above) we propose to sample the components transition times from a joint distribution uniform on the support defined by the scenario selected (Turati et al., 2015). For the sake of clarity, let us consider a scenario S_j that involves the change of states of two components, namely, *A* and *B,* in a specific order within the mission time (e.g., in this case $T_A < T_B \leq T_{Miss}$). Then, in order to simulate one possible system evolution, we sample the couple (T_A, T_B) from the joint uniform distribution on the support defined by S_j, which is shown in Fig. 2 as a shaded region. For this purpose, we resort to a MCMC Gibbs sampling (Robert & Casella, 2004).

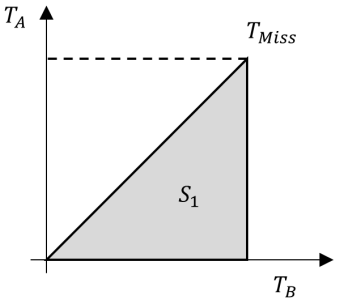

Fig. 2 Support of the time of occurrence of events *A* and *B* defining scenario S_j, where $T_A < T_B \leq T_{Miss}$.

By resorting to a uniform distribution, a more thorough exploration of the time window under analysis is obtained. Actually, the occurrences of component failures are frequently modeled by exponential distributions; this implies that the likelihood of the time of occurrence of an event decreases with the increment of its value. This makes some system evolutions and, hence, some ESs extremely unlikely to occur, thus reducing the exploration capability of a method based on plain random sampling from those (original) distributions. In this respect, Fig. 3 shows the probability density function of the time T_{int} of the initiating event introduced in the trivial example above, in the cases of an exponential conditional distribution with support defined on $[0, T_{miss} = 100]$ and of a uniform distribution defined on the same support.

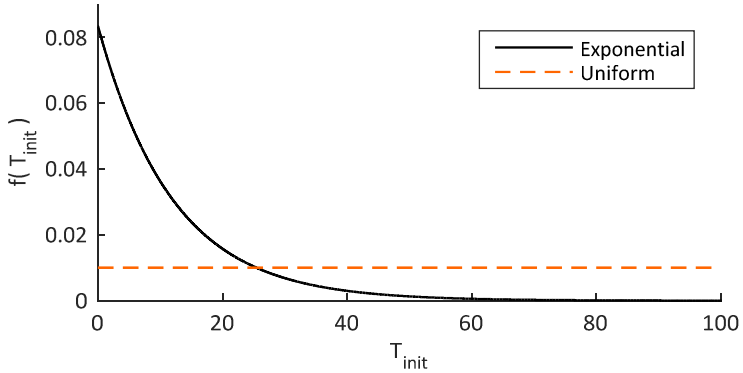

Fig. 3 Exponential pdf (dark-line) and uniform pdf (light-dashed line) defined on the support $[\mathbf{0}, \boldsymbol{T_{Miss}}$=100]

3 A Novel Framework For The Semi-Automatic Adaptive Exploration Of The State Space Of Dynamic Systems

The Semi-Automatic Adaptive Exploration method consists of three main steps, as sketched in the flow diagram of Fig. 4:

1) preliminary exploration (Section 3.1), i.e., a global exploration of the whole space of the dynamic system according to two possible driving criteria, namely, the *guided* (Section 3.1.1) and the *forced* (Section 3.1.2) exploration; this step aims at enhancing the general knowledge of the analyst regarding the role and impact of different time sequences in the evolution of the dynamic system;

2) interactive decision making (Section 3.2), i.e., after the preliminary exploration, the analyst can decide to either improve his/her global view of the state space by increasing the number of simulations according to the criteria of the *preliminary exploration* (step 1), or focus his/her attention on a specific event of interest (step 3);

3) deep exploration (Section 3.3), i.e., a thorough exploration of a particular event: for example, the objective can be that of retrieving the possible evolutions within the scenario S_j that can reach a given End State ES_i, indicated hereafter as the pair $\{S_j, ES_i\}$.

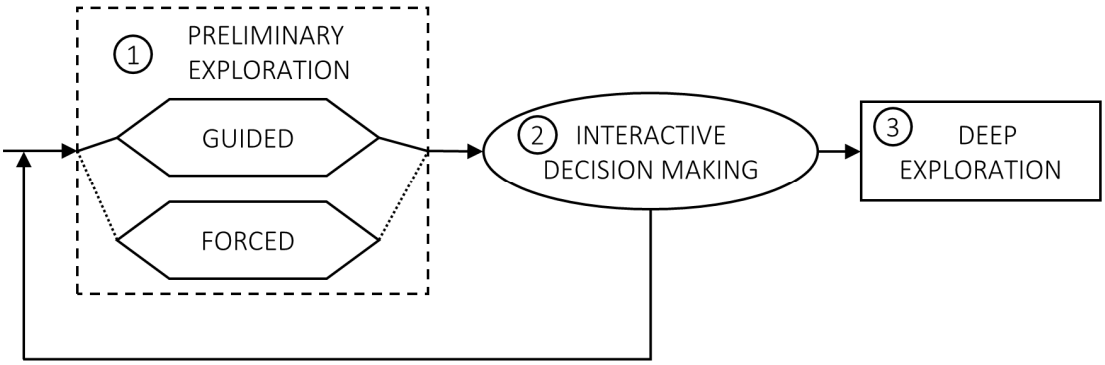

Fig. 4 General flow diagram of the exploration framework proposed

3.1 Preliminary Exploration

The preliminary exploration aims at efficiently retrieving information about the general dynamic behavior of the system (model) under the constraint of limited computational effort (i.e., of a fixed number of available simulations to run): in other words, for each scenario S_j, it aims at identifying all the ESs that it can "generate". For this purpose, the analyst can choose either a *preliminary guided exploration* (Section 3.1.1) or a *preliminary forced exploration* (Section 3.1.2) according to his/her interest and to the information already available.

3.1.1 Guided Exploration

If the analyst does not have any prior information about the outcomes of the system accident scenarios or if he/she has scarce information, but he/she is not interested in any specific outcome based on the current knowledge at his/her disposal, a "guided exploration" framework should be chosen. In particular, the choice of the scenario that will be explored through a new simulation run is automatically made by selecting the scenario S^* which maximizes the driving function $I_\gamma(\cdot)$:

$$S^* = \underset{j \in S}{\operatorname{argmax}}\, I_\gamma(S_j), \tag{1}$$

where $I_\gamma(S_j)$ is defined as:

$$I_\gamma(S_j) = I_\gamma(n_j,\, N_j^{ES}) = \frac{(N_j^{ES})^\gamma}{n_j}, \tag{2}$$

where N_j^{ES} is the number of ESs that Scenario S_j can reach (if this information is not available, then it represents the number of ESs that have already been visited within the scenario and it is updated when a new ES is discovered by a new simulation run); n_j is the number of simulations that have already been run within S_j; $\gamma \in (-\infty, +\infty)$ is a parameter that should be chosen according to

the preference of the analyst: (*i*) if γ<0, the driving function chooses more frequently those scenarios that can reach a small number of ESs; (*ii*) if $\gamma = 0$, no preference is given to any scenario; (*iii*) if $\gamma > 0$, the driving function selects more likely those scenarios that can reach a large number of ESs. For the sake of clarity and by way of example, consider a simple dynamic system where only four scenarios can occur $S_1, \dots, S_4$ and where each scenario can reach a different number of end states, $N_1^{ES} = 1, N_2^{ES} = 2, \dots, N_4^{ES} = 4$. Finally, let assume that all reachable ESs in the same scenario have the same probability of occurring. Table I reports the average of 1000 explorations, performed with 100 simulations each, that have been distributed among the different scenarios according to three different values of the parameter γ, i.e., $\gamma = -1$ (left), $\gamma = 0$ (middle) and $\gamma = 1$ (right).

Table I Average results of 1000 experiments. Each experiment runs 100 simulations of preliminary guided exploration with different values of the parameter γ: -1 (left); 0 (middle); 1 (right). Column "Tot" represents the total number of simulations run within the respective scenario.

	ES_1	ES_2	ES_3	ES_4	TOT	ES_1	ES_2	ES_3	ES_4	TOT	ES_1	ES_2	ES_3	ES_4	TOT
S_1	47.9	0.0	0.0	0.0	**47.9**	25.0	0.0	0.0	0.0	**25.0**	10.0	0.0	0.0	0.0	**10.0**
S_2	12.0	11.9	0.0	0.0	**23.9**	12.4	12.6	0.0	0.0	**25.0**	10.0	9.9	0.0	0.0	**20.0**
S_3	5.3	5.3	5.3	0.0	**15.9**	8.4	8.3	8.3	0.0	**25.0**	10.0	10.0	10.0	0.0	**30.0**
S_4	3.0	3.1	3.0	3.1	**12.2**	6.2	6.2	6.3	6.3	**25.0**	10.0	10.0	9.9	10.1	**40.0**

The choice of parameter $\gamma = 1$ is particularly suitable because, in this case, the exploration algorithm distributes the simulations among all the scenarios in order to guarantee that each scenario S_j "gathers" a number of simulations proportional to the number N_j^{ES} of end states that each scenario can "generate".

3.1.2 Forced Exploration

As in the preliminary guided exploration, the algorithm selects the scenario S^* which maximizes a given driving function and, then, simulates a dynamic evolution conditioned to the selected scenario S^*. However, the driving function, namely $I_\beta(\cdot)$, of the preliminary forced exploration is defined as:

$$I_\beta(S_j, ES^*) = I_\beta(n_j, N_j^{ES}, I_{ES^*}) = \begin{cases} \dfrac{(N_j^{ES})^\gamma}{n_j}, & I_{ES^*} = 0 \\ \dfrac{(N_j^{ES})^\gamma}{n_j} \cdot \beta, & I_{ES^*} = 1 \end{cases}, \tag{3}$$

where ES^* is a particular set of end states of interest for the analyst; I_{ES^*} is a Boolean variable, which equals 1 if the simulations of scenario S_j can reach at least one of the end states in ES^*, and 0 otherwise; $\beta \in (0, +\infty)$ is the forcing parameter: the higher it is, the more frequently the

algorithm selects those scenarios that can reach the end states in ES^*; finally, the remaining variables are defined as in Eq. (2). With respect to the case of the preliminary exploration, a different number of ESs can be included in the set ES^*, i.e., the cardinality of ES^* can be larger than 1.

For the sake of clarity, let us consider the same trivial example introduced above (Table I), assume parameter $\gamma = 1$ and suppose that we are interested in gathering information about those scenarios that can reach end states $ES^* = \{ES_3; ES_4\}$ (for example, because they represent extremely dangerous conditions). Table II reports the effects of different choices of parameter $\beta = \{0.25;\ 1;\ 4\}$ on the final distribution of the simulation runs among the scenarios. If $\beta \in (0,1)$, those scenarios that can reach the set ES^* are penalized in the selection step (left); if $\beta = 1$, the algorithm turns to the preliminary guided exploration described above (middle); finally, if $\beta \in (1, +\infty)$, the scenarios that can reach the set ES^* are favored in the selection step (right).

Table II Average results of 1000 experiments. Each experiment run 100 simulations of preliminary forced exploration with $\gamma = 1$ and with different values of the parameter β: 0.25 (left); 1 (middle); 4 (right). Column "Tot" represents the total number of simulations run within the respective scenario.

	ES_1	ES_2	ES_3	ES_4	TOT	ES_1	ES_2	ES_3	ES_4	TOT	ES_1	ES_2	ES_3	ES_4	TOT
S_1	10.0	0.0	0.0	0.0	**10.0**	7.0	0.0	0.0	0.0	**7.0**	3.1	0.0	0.0	0.0	**3.1**
S_2	10.0	9.9	0.0	0.0	**20.0**	7.0	6.0	0.0	0.0	**12.9**	3.0	2.9	0.0	0.0	**5.9**
S_3	10.0	10.0	10.0	0.0	**30.0**	12.5	11.7	11.7	0.0	**36.0**	12.6	12.3	12.5	0.0	**37.4**
S_4	10.0	10.0	9.9	10.1	**40.0**	12.7	11.8	11.8	11.7	**48.1**	13.4	13.4	13.4	13.3	**53.5**

For the preliminary forced exploration, we have proposed only one function based on one single parameter beta, which reflects an interest about a set of known end states; however, a variety of functions could be used at this stage to force the selection of scenarios according to other desirable criteria.

3.2 Interactive Decision Making

Every time a preliminary exploration is performed, matrixes, such those reported in Table I and Table II, become available. Hence, based on the events visited (i.e., on the pairs Scenario-End State $\{S_j, ES_i\}$) and on the number of simulations that have been run to visit them, the analyst can decide either to increase the number of simulations according to the criteria adopted in the preliminary exploration phase or to perform a deeper and more refined exploration of specific events of interest. According to his/her preference, the analyst has to iteratively choose the maximum allowable number of simulations that can be run according to the *preliminary* or *deep* exploration, respectively. In many cases, the dimension of the system (state space) and the variability of its behavior (in practice, the number of ESs a scenario can reach and the corresponding probabilities),

are not known a priori; on the contrary, the computational cost needed for a system simulation can be known (e.g., in terms of average time per simulation). Then, the computational effort can be considered as a constraint that the analyst needs to take into account in accordance with his/her preferences among the different exploration criteria. In this respect, it must be noticed that the proposed method does not guarantee that the whole event space is probed: inevitably, if the computational capacity available (in practice, the total number of simulations that can be run) is small compared to the size of the system state space, only a limited number of end states can be explored for each scenario.

3.3 Deep Exploration

The objective of the deep exploration is to identify as precisely as possible, which system evolutions (i.e., in practice which combinations of transition times) can lead to a given event of interest. For the sake of clarity, we assume that an event of interest is defined as the pair {Scenario-End-State} = $\{S_j, ES^*\}$; nonetheless, with no loss of generality ES^* can represent also an ensemble of end states. Given the structure of the mathematical model, the guiding idea of the deep exploration is to generate time sequences "around" those that have already reached the event $\{S_j, ES^*\}$. In order to achieve this goal, we resort to a Markov Chain Monte Carlo (MCMC) method, which allows to generate a set of random samples from any desired (namely, *target*) probability distribution $p(\cdot)$ (Robert & Casella, 2004). In detail, we utilize a Metropolis-Hasting (M-H) algorithm (Chib & Greenberg, 1995) to sample components transition times uniformly on the support *SES** of the event of interest $\{S_j, ES^*\}$, in other words, to sample uniformly among the transition times that lead to the event of interest $\{S_j, ES^*\}$. The Metropolis-Hasting algorithm consists of two steps: *i*) proposition of a new candidate $\boldsymbol{T}^*$ (in this case, a vector of transition times) in accordance to a proposal distribution $q(\cdot)$; *ii*) acceptance or rejection of the proposed time vector. For the first step, we utilize as proposal a Multivariate Gaussian distribution $q(\boldsymbol{T}^* \mid \boldsymbol{T_n}) \sim N(\boldsymbol{T_n}, \boldsymbol{\Sigma})$, having as mean value the last vector of transition times $\boldsymbol{T_n}$ accepted in the region of interest and as covariance matrix $\boldsymbol{\Sigma}$. In particular, $\boldsymbol{\Sigma}$ is estimated using the set of transition times generated during the preliminary exploration that have led the system to the event of interest. This choice of the covariance matrix provides a "favorable" Acceptance Ratio (*AR*) between the number of proposed and accepted values. Regarding the second step, the proposed value $\boldsymbol{T}^*$ is accepted (i.e., $\boldsymbol{T_{n+1}} = \boldsymbol{T}^*$) or rejected (i.e., $\boldsymbol{T_{n+1}} = \boldsymbol{T_n}$) with a probability $\alpha(\boldsymbol{T_n}, \boldsymbol{T}^*) = \min(r(\boldsymbol{T_n}, \boldsymbol{T}^*), 1)$, where *r* is defined as follows:

$$r(\boldsymbol{T_n}, \boldsymbol{T}^*) = \begin{cases} \dfrac{p(\boldsymbol{T}^*) \cdot q(\boldsymbol{T_n} \mid \boldsymbol{T}^*)}{p(\boldsymbol{T_n}) \cdot q(\boldsymbol{T}^* \mid \boldsymbol{T_n})}, & p(\boldsymbol{T_n}) \cdot q(\boldsymbol{T}^* \mid \boldsymbol{T_n}) > 0 \\ 1, & otherwise \end{cases} \quad (4)$$

$p(\cdot)$ being the *target* distribution from which we want to sample. Thanks to the symmetry of the Gaussian proposal $q(\boldsymbol{T_n} \mid \boldsymbol{T}^*) = q(\boldsymbol{T}^* \mid \boldsymbol{T_n})$, then, Eq. (4) can be rewritten as:

$$r(\boldsymbol{T_n}, \boldsymbol{T}^*) = \begin{cases} \dfrac{p(\boldsymbol{T}^*)}{p(\boldsymbol{T_n})}, & p(\boldsymbol{T_n}) > 0 \\ 1 \quad , & otherwise. \end{cases} \tag{5}$$

Finally, since the target distribution is uniform on the support SES^* of the event of interest, then, the probability $\alpha(\boldsymbol{T_n}, \boldsymbol{T}^*)$ can be written as:

$$\alpha(\boldsymbol{T_n}, \boldsymbol{T}^*) = \begin{cases} 1, & \boldsymbol{T}^* \in SES^* \\ 0, & otherwise. \end{cases} \tag{6}$$

However, in order to know whether the proposed occurrence time vector $\boldsymbol{T}^*$ belongs to SES^*, we need to verify that: *a*) $\boldsymbol{T}^*$ satisfies the time order characterizing scenario S_j; *b*) $\boldsymbol{T}^*$ leads to the end state of interest ES^*. It is worth verifying the two conditions in the mentioned order, since the second condition implies a system simulation run and the corresponding computational cost.

It must be kept in mind that, whenever the M-H algorithm is used, the Acceptance Ratio (*AR*) between the proposed samples and the accepted ones plays a fundamental role. Too high acceptance ratios (*AR* > 0.9) are a symptom of a proposal $q(\cdot)$ with too small variability, i.e. most of the proposed $\boldsymbol{T}^*$ are too close to the original ones and, thus, the algorithm results too slow in probing the support SES^*; on the contrary, too small acceptance ratios (*AR* < 0.2) are a symptom of a proposal $q(\cdot)$ with too high variability, i.e., most of the proposed $\boldsymbol{T}^*$ are likely to fall out of the support of interest SES^*. In this respect, adaptive MCMC methods exploiting an adaptive proposal distribution have been presented in the literature and can be employed at this stage to "optimally" fill the support SES^* of interest (Andrieu & Thoms, 2008; Roberts & Rosenthal, 2009). Nonetheless, if the majority of the proposed samples are rejected due to the constraint (*a*) of belonging to a given scenario (i.e., of presenting a given precise event timing), an MCMC Gibbs Sampler (G-S), such as the one introduced in Section 2, may be useful. Actually, the G-S allows sampling occurrence time vectors that implicitly satisfy the time order constraint (*a*), probing the *entire* scenario instead of the support SES^* *only*. Thus, roughly speaking, on the basis of the "area" of the support of the scenario S_j that is covered by the end state $ES^*$of interest, the analyst can choose different MCMC methods: if the portion of transition time vectors that leads to ES^* is small (Fig. 5, right), an M-H algorithm should be preferred to fill this "small portion" of scenario S_j; on the contrary, if the portion of transition times vector leading to ES^* is large (Fig. 5, left), G-S should be preferred, in order to put more effort on the "even coverage" of the entire scenario S_j.

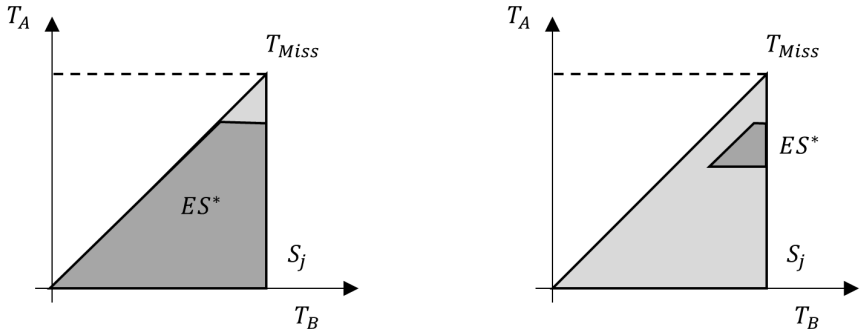

Fig. 5 The dark area represents the portion of scenario S_j that leads to the end state of interest ES^*.

Regarding the approach used to choose the number of simulations to run for performing the deep exploration, two criteria are proposed: (*i*) fixed number of simulations (as in the *preliminary exploration* of Section 3.1); (*ii*) level of filling of the support of the event of interest. For what concerns the second criterion, the idea is to keep on generating new simulation outcomes until the support of the event of interest is filled by an amount of points (i.e., configurations) that "sufficiently" cover the entire outcome variability. In detail, after the preliminary exploration a set of occurrence time vectors $EX_V(SES^*) = \{\boldsymbol{T_1}, \dots, \boldsymbol{T_V}\}$ that lead to the event of interest $\{S_j, ES^*\}$ is available. As a measure of the (time) space filling, the maximum of the minimum distances among these time vectors is considered: then, a time filling index $D_V\big(EX_V(SES^*)\big)$ after the preliminary exploration is computed as:

$$D_V\big(EX_V(SES^*)\big) = \max_{i \in EX_V(SES^*)} \min_{j \neq i} d\big(\boldsymbol{T}_i, \boldsymbol{T}_j\big) \tag{7}$$

where $d(\cdot,\cdot)$ represents a proper distance between two vectors. In this paper, for example, we consider the Euclidean one. Whenever a new time vector $\boldsymbol{T}_n$ is accepted during the exploration, it is added to the set of time vectors that lead to the event of interest, i.e., $EX_n(SES^*) = \{EX_{n-1}(SES^*); \boldsymbol{T}_n\}$, and the filling index $D_n\ \big(EX_n(SES^*)\big)$ is consequently updated. The deep exploration ends when the ratio between the current filling index and the preliminary one falls below a fixed threshold $\delta \in [0,1]$, i.e., when the "density" of time vectors in the support *SES** of interest is $\sim(1/\delta)^l$ times higher than the preliminary one, being l the size of the time vector $\boldsymbol{T}_n$. Nonetheless, a maximum allowable number n_{max} of samples is also set, in order to limit in any case the maximum computational effort. Then, the stopping criterion becomes:

$$\frac{D_n\big(EX_n(SES^*)\big)}{D_V\big(EX_V(SES^*)\big)} < \delta \ or \ n > n_{max}\ . \tag{8}$$

The corresponding algorithm is summarized in Table III.

Table III Sketch of the algorithm describing the deep exploration stopping criterion.

1. For $i = 1, \dots, V$ evaluate the minimum distances from the vector $\boldsymbol{T}_i$ and save them in the vector $\boldsymbol{d}_V$:
   $d_V(i) = \min_{j \neq i} d(\boldsymbol{T}_i, \boldsymbol{T}_j)$.
   According to this notation $D_V(EX_V(SES^*)) = \max \boldsymbol{d}_V$.
2. Given a new time vector $\boldsymbol{T}_n$, update the $\boldsymbol{d}_{n-1}$vector for $i = 1, \dots, n-1$:
   $d_n(i) = \min(d_{n-1}(i), d(\boldsymbol{T}_i, \boldsymbol{T}_n))$,
3. Add the *n*-th component to $\boldsymbol{d}_{n-1}$ resorting to the distance already available from the previous step:
   $d_n(n) = \min_{j \neq n} d(\boldsymbol{T}_n, \boldsymbol{T}_j)$.
4. Evaluate the filling index:
   $D_n(EX_n(SES^*)) = \max \boldsymbol{d}_n$.
5. Check if the stopping criteria are satisfied:
   $\frac{D_n(EX_n(SES^*))}{D_V(EX_V(SES^*))} < \delta \; or \; n > n_{max}$
   If not, return to step 2.

The space filling capability of the algorithm is strictly related to the dimension of the vectors involved: in practice, the higher the dimension, the larger the number of random vectors needed to reduce the filling index.

4 Case Study

The case study under analysis is a gas transmission subnetwork composed of two pipes in parallel and another one in series. The input of each pipe is controlled by a valve. The whole block diagram is shown in Fig. 6, where each pair valve-pipe is considered as a single block.

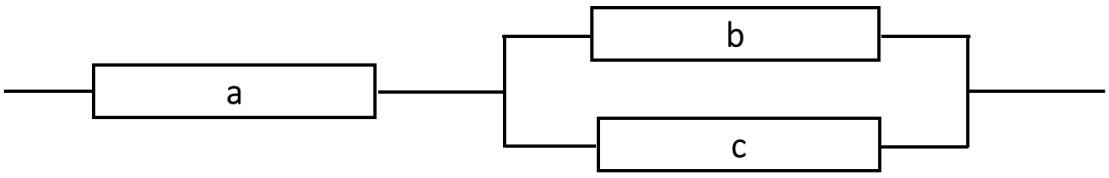

Fig. 6 Block diagram of the system under analysis.

Each pipe can transmit gas with a maximum flow rate of $[\phi_a, \phi_b, \phi_c] = [8,5,5] \cdot 10^4 \; m^3/day$, for pipes *a, b, c,* respectively. A control system adjusts the opening of the valves in order to guarantee the equilibrium between the input and output flows. Fig. 7 shows the event tree containing all the scenarios that can occur in the system. If one of the pipes in parallel breaks, the control system immediately closes the corresponding valve and increases the flow rate of the remaining pipe to the maximum, in order to compensate for the diminished flow. No reparation strategies are considered. The system presents 8 possible scenarios with different operating

conditions: *i) safe*, i.e., all pipes are functioning correctly; *ii) overloaded*, i.e., one of the pipes in parallel is closed; *iii) broken*, i.e., no gas is provided by the system.

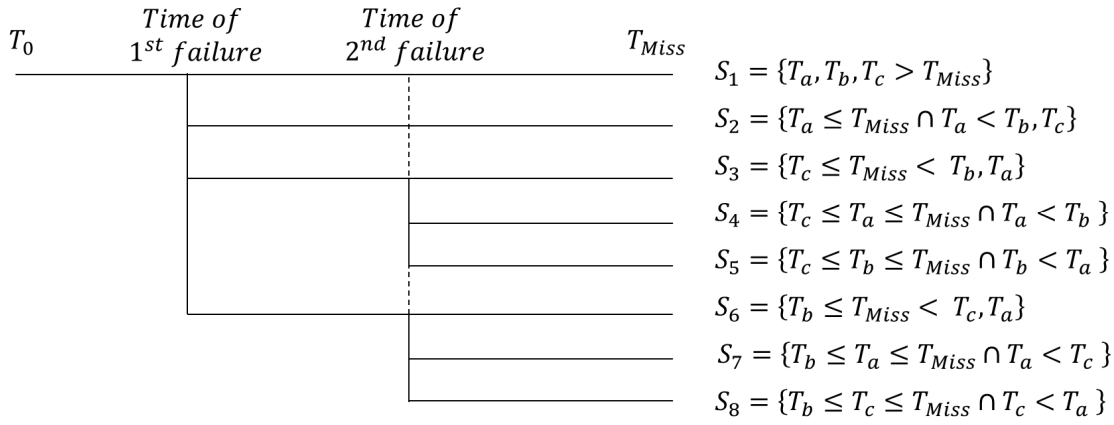

Fig. 7 Event tree representation of the 8 scenarios that can occur, where T_a, T_b, T_c are the times of failures of components *a*, *b*, *c*, respectively, and T_{Miss} is the mission time.

The ESs for each scenario have been defined and classified on the basis of two output variables: *i)* the amount of Gas provided in Safe Conditions (GSC), i.e., when all the components are functioning correctly; *ii)* the amount of Gas provided in Overloaded Conditions (GOC), i.e., when one of the two pipes in parallel is down and the remaining one works at its maximum flow rate. With respect to that, GSC_{max} and GOC_{max} indicate the maximum quantities of gas that can be provided within the mission time $T_{Miss} = 900d$, in safe and overloaded conditions, respectively, i.e., $GSC_{max} = \phi_a \cdot T_{Miss}$ and $GOC_{max} = \max(\phi_b, \phi_c) \cdot T_{Miss}$. The ESs are, then, divided into six classes according to the criteria reported in Fig. 8. For example, $ES_4 = \left\{\frac{1}{3}GSC_{max} < GSC \leq \frac{2}{3}GSC_{max} \cap 0 \leq GOC \leq \frac{1}{3}GOC_{max}\right\}$, which means that the system has operated for a medium period of time in *safe* conditions $\left(\frac{1}{3}GSC_{max} < GSC \leq \frac{2}{3}GSC_{max}\right)$ and, then, once it goes in *overloaded* conditions, it breaks down $\left(0 \leq GOC \leq \frac{1}{3}GOC_{max}\right)$.

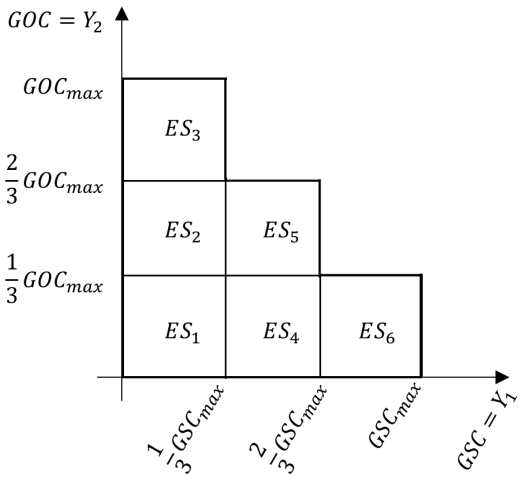

Fig. 8 Classification of the End States (ESs) according to the 2 output variables GSC and GOC

It must be noticed that not all the ESs can be reached by all scenarios: Table IV (left matrix) reports those ESs that can be reached by a given scenario (indicated by 1) and those that cannot (indicated by 0): each column in the Table represents an ES and each row represents a scenario. This information is usually not available a priori and in general its retrieval represents one of the objectives of the state space exploration. However, it is used here to analyze the performance of the proposed method. In Table IV (middle and right), two additional matrices show the ESs reachable for two sets of different gas flow rates, e.g., $[\phi_a, \phi_b, \phi_c] = [8, 3.7, 5] \cdot 10^4\ m^3/day$ and $[\phi_a, \phi_b, \phi_c] = [8, 2.2, 6] \cdot 10^4\ m^3/day$, respectively. These values have been chosen in order to analyze the performance of the method for different parameters values, which imply that the number of reachable ESs varies.

Table IV Matrices of the end states that the system can reach for each scenario for different sets of flow rate parameters values: $[\boldsymbol{\phi_a, \phi_b, \phi_c}] = \boldsymbol{[8, 5, 5] \cdot 10^4\ m^3/day}$ (left); $[\boldsymbol{\phi_a, \phi_b, \phi_c}] = \boldsymbol{[8, 3.7, 5] \cdot 10^4\ m^3/day}$ (middle) and $[\boldsymbol{\phi_a, \phi_b, \phi_c}] = \boldsymbol{[8, 2.2, 6] \cdot 10^4\ m^3/day}$ (right).

	ES_1	ES_2	ES_3	ES_4	ES_5	ES_6	ES_1	ES_2	ES_3	ES_4	ES_5	ES_6	ES_1	ES_2	ES_3	ES_4	ES_5	ES_6
S_1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1
S_2	1	0	0	1	0	1	1	0	0	1	0	1	1	0	0	1	0	1
S_3	0	0	1	0	1	1	0	1	1	1	1	1	0	1	0	1	1	1
S_4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1
S_5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1
S_6	0	0	1	0	1	1	0	0	1	0	1	1	0	0	1	0	1	1
S_7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
S_8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

4.1 Preliminary Guided Exploration

To evaluate the performance of the *preliminary guided exploration*, two indices are introduced: (*i*) the Number of simulations needed for the First complete Exploration (*NFE*), i.e., the number of simulations that should be run to visit at least once all the reachable ESs for all the scenarios; (*ii*) the Number of simulations needed for the Second complete Exploration (*NSE*), i.e., the number of simulations that should be run to visit all the reachable ESs for all the scenarios at least twice. *NFE* gives information about the number of simulations needed to explore all the events defined by the pairs {Scenario, End-State} = {S,ES}, when the matrices shown in Table IV (i.e., the ESs) are not known yet. On the contrary, *NSE* gives information about how the simulations are efficiently distributed among the different scenarios, once the matrices in Table IV (i.e., the ESs) begin to be known as a result of the preliminary exploration. The results of the preliminary guided exploration with $\gamma = 1$ are compared to those of: 1) a crude Monte Carlo simulation method (MC), that randomly selects the scenario and, then, simulates the proper transition times according to the uniform sampling criterion proposed in Section 2; 2) an entropy-driven exploration(Turati et al., 2015), which follows a procedure similar to the *guided exploration*, but with an entropy-driven function instead of $I_\gamma(\cdot)$.

Fig. 9 - Fig. 11 show the empirical cumulative density functions (cdfs) of NFE (left) and NSE (right) built on 1000 samples for the three configurations of the parameters considered in Table IV. It can be seen that the preliminary guided exploration with $\gamma = 1$ achieves better or at least comparable performance than the entropy–driven exploration in all configurations considered, for both indexes. This is shown by the fact that the cdfs obtained by the *guided* exploration (light dash line) are "shifted" to the left with respect to the cdfs obtained by the entropy-driven method (dark dotted line). Both these two steered methods (γ-guided and entropy-driven) outperform the crude MC (light line) in the majority of the configurations of the parameters considered. However, the performance of the *guided* and of the entropy-driven explorations could be worse than those of the crude MC method for some configurations of the parameters (see, e.g., Fig. 11). Indeed, if the configuration is such that the rarest event to visit is inside a scenario (e.g., S_4, S_5 in this case) that may reach a small number of ESs (i.e., only 4), then, using a method which increases the number of simulations in those scenarios that can reach a larger number of ESs is not effective. Nonetheless, while the entropy-driven method is stuck, the *guided* exploration allows changing parameter γ in order to increase the exploration effectiveness.

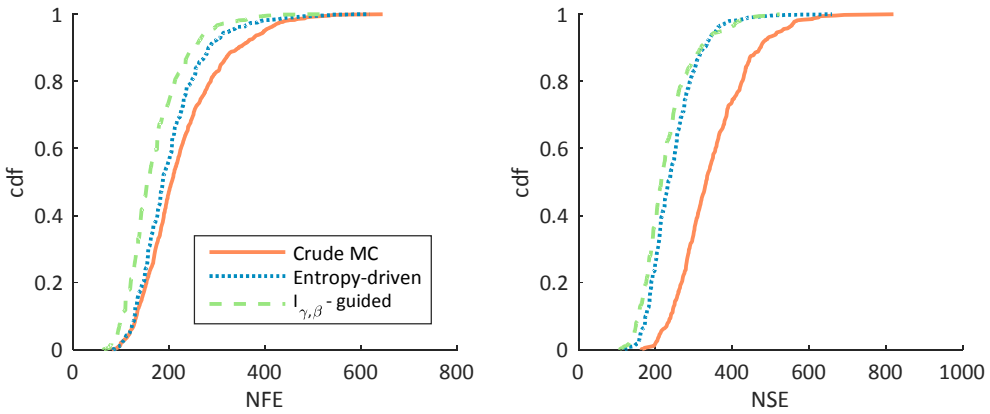

Fig. 9 Empirical cdfs of the NFE (left) and of the NSE (right) for crude MC (light line), for an entropy-driven method (dark dotted line) and for the preliminary guided exploration with $\boldsymbol{\gamma = 1}$ (light dashed line) with flow rate parameters $[\boldsymbol{\phi_a, \phi_b, \phi_c}] = \boldsymbol{[8, 5, 5] \cdot 10^4\ m^3/day}$.

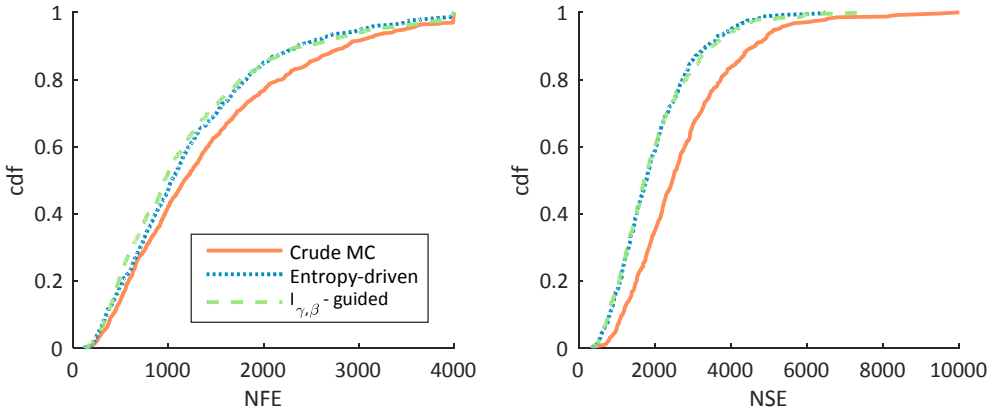

Fig. 10 Empirical cdfs of the NFE (left) and of the NSE (right) for crude MC (light line), for an entropy-driven method (dark dotted line) and for the preliminary guided exploration with $\boldsymbol{\gamma = 1}$ (light dashed line) with flow rate parameters $[\boldsymbol{\phi_a, \phi_b, \phi_c}] = \boldsymbol{[8, 3.7, 5] \cdot 10^4\ m^3/day}$.

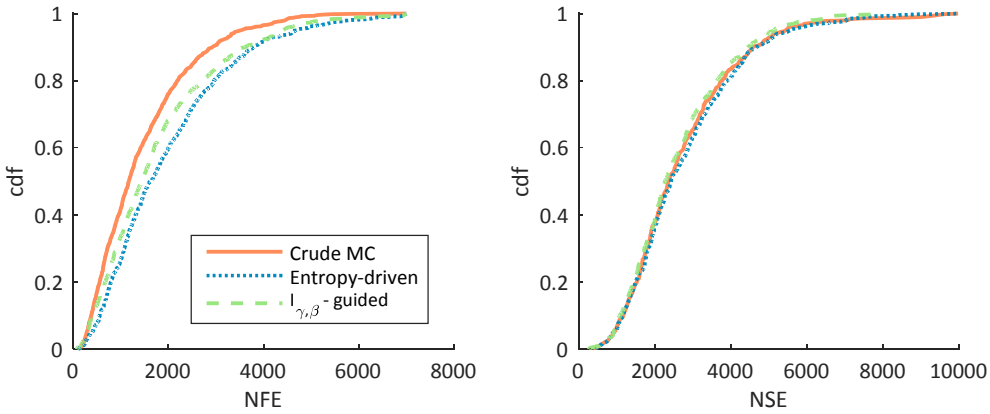

Fig. 11 Empirical cdfs of the NFE (left) and of the NSE (right) for crude MC (light line), for an entropy-driven method (dark dotted line) and for the preliminary guided exploration with $\boldsymbol{\gamma = 1}$ (light dashed line) with flow rate parameters $[\boldsymbol{\phi_a, \phi_b, \phi_c}] = \boldsymbol{[8, 2.2, 6] \cdot 10^4\ m^3/day}$.

4.2 Preliminary Forced Exploration

For the analysis of the effects of the *preliminary forced exploration,* we have chosen the system configuration with flow rate parameter vector $[\phi_a, \phi_b, \phi_c] = [8, 2.2, 6] \cdot 10^4\ m^3/day$. In this setting, we assume to be interested in thoroughly exploring those scenarios that can lead to ES_3. The performance of the *forced* exploration is assessed by means of the average percent increment of simulations falling inside the scenarios of interest, i.e., those that can reach ES_3, with respect to the preliminary *guided* exploration. Different values of the parameter $\beta = \{2;\ 4;\ 8\}$ are considered to show the effects on the increment in the presence of different levels of computational resources available, i.e., with respect to different numbers of simulation runs $[250; 500; 1000; 2000; 4000]$. Fig. 12 reports the average percent increment and the respective 0.1 and 0.9 percentiles on 1000 experiments for each combination of the values of parameter β and number of simulations, for a chosen scenario of interest (S_7). According to the definition of the function in Eq. (3), the larger the parameter β, the larger the average percentage increment of simulations in the scenario of interest, i.e., around (35, 60, 80)% for $\beta = \{2;\ 4;\ 8\}$, respectively. However, it must be considered that if parameter β is too large compared to the computational effort, then the performance can be more uncertain, as shown by the intervals corresponding to $\beta = 8$ and a number of simulations lower than 500. This is due to the fact that if the parameter β is too large, then the *forced* exploration will focus its attention (i.e., the simulations) on the scenario that firstly discovers (i.e., reaches) the end state of interest (e.g., ES_3) before other scenarios can reach the end state of interest (ES_3) as well. The larger the number of scenarios that can reach the *ES* of interest, the larger the sensitivity to the number of simulations, given β. We do not report the Figures for the other two scenarios of interest (S_6 and S_8), since they show similar results.

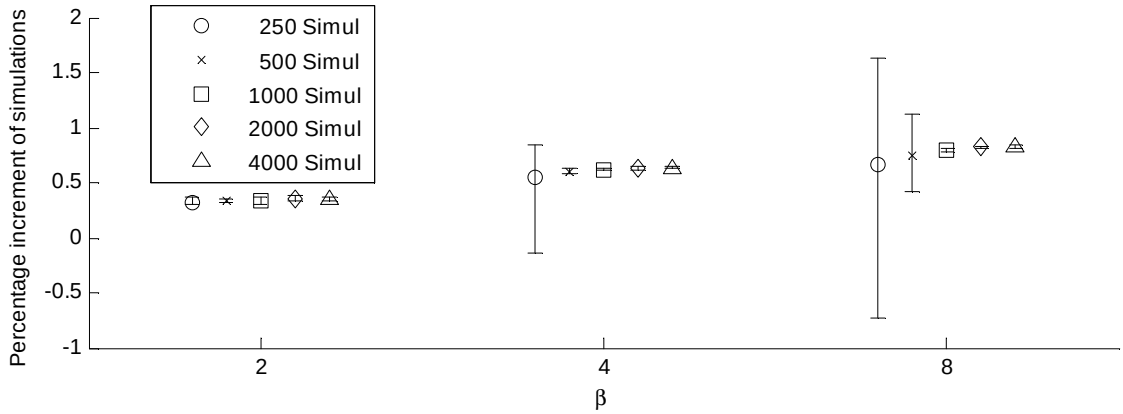

Fig. 12 Average percentage increment of simulations in a given scenario of interest S_7 and the respective percentile (10% and 90%), for parameter $\beta = \{2;\ 4;\ 8\}$ and for different numbers of simulations.

It must be pointed out that the average increment of simulations in the scenarios of interest is conditioned to the number of scenarios that can reach the end state of interest. Indeed, assuming

that we are interested in exploring end state ES_6 (which can be reached by all scenarios), then no preference is given to any scenario since the *forcing* function equally increments the preference for all of them. Furthermore, since the information about the number of end states that a given scenario can reach is typically not available, the previous observation suggests that the *forced* scheme guarantees an exploration at least as effective as the *guided* one, when all scenarios can reach the end state of interest.

4.3 Deep Exploration

After a preliminary guided exploration of the system defined by parameters $[\phi_a, \phi_b, \phi_c] = [8,3.6,5] \cdot 10^4\ m^3/day$ a large variability in the outcomes is observed within scenario S_5, as highlighted in Table V. Thus, it is interesting to retrieve the event time sequences that lead to two chosen ESs: ES_1, which represents the worst final condition, and ES_3, which has been visited only few times during the preliminary exploration (e.g. according to Table V, highlighted row).

Table V Matrix reporting the ESs visited by a preliminary guided exploration of the system with parameters $[\boldsymbol{\phi_a}, \boldsymbol{\phi_b}, \boldsymbol{\phi_c}] = [\mathbf{8}, \mathbf{3.6}, \mathbf{5}] \cdot \mathbf{10^4}\ \boldsymbol{m^3/day}$, given a computational effort of 1000 simulations.

	ES_1	ES_2	ES_3	ES_4	ES_5	ES_6
S_1	0	0	0	0	0	29
S_2	21	0	0	38	0	28
S_3	0	27	10	24	36	47
S_4	46	29	0	41	5	23
S_5	39	50	2	57	7	18
S_6	0	0	23	0	28	36
S_7	38	36	22	36	14	26
S_8	34	39	24	41	12	22

The space filling parameter is set to 0.2 with a maximum number of simulations to run set to 5000. The covariance matrix has been estimated from the vectors of transient times obtained from the preliminary exploration with respect to ES_1. On the contrary, since only two vectors are available for ES_3, an independent Gaussian proposal with standard deviation equal to the Euclidean distance between the two vectors is considered. The chosen standard deviation provides an idea of the dimension of the support to explore. Fig. 13 reports the transition time vectors of the scenario of interest S_5 after the preliminary exploration (on the left) and after the deep exploration (on the right). Results confirm that the proposed deep exploration is capable of increasing the number of simulations around the time sequences that reach the ES of interest, increasing the information about those sequences that can lead the system to a particular event. For example, in order to

obtain ES_3, pipe *c* should break within the initial 100 days whereas pipe *b* should work at least for 800 days after the failure of the first one.

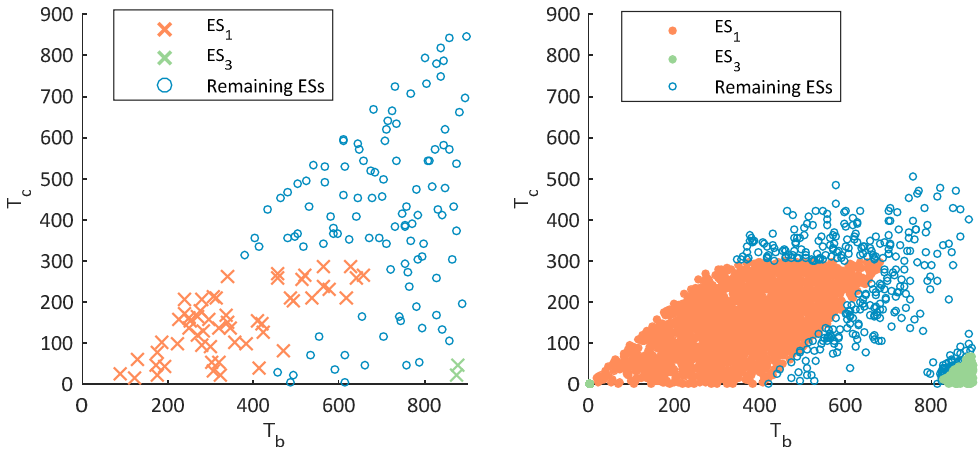

Fig. 13 Preliminary guided exploration of S_5 (left) and deep exploration of ES_1 and ES_3 in the same scenario (right).

5 Conclusions

IDPSA is expected to lead to a more realistic evaluation of the risks associated to safety-critical systems (e.g., nuclear power plants). One opportunity that is sought is the discovery and understanding of the possible outcomes from the dynamics of accidents, leaving out as little as possible of unexpected. Thorough system state space exploration in IDPSA allows identifying extreme situations and critical system characteristics for preventing accidents and/or mitigating their effects.

In this paper, a novel framework for the exploration of the system state space has been proposed. The framework allows including analyst prior knowledge about the system and focusing the exploration on particular configurations of interest, e.g., due to their criticality and/or rarity. Three methods, namely, *guided, forced* and *deep exploration,* are implemented to allow diversifying the exploration in accordance with the preferences and interests of the analyst. Application to a simple system has shown that the proposed framework outperforms an entropy-driven method of literature (Turati et al., 2015), as well as a crude Monte Carlo method, in exploring the system state space. In addition, the deep exploration phase has been shown capable of leading a large number of simulations to the events of interest.

Future research efforts will be devoted to extending the exploration framework to the assessment of the probability of the events of interest and to applications to more complex case studies.

References

Aldemir, T. (2013). A survey of dynamic methodologies for probabilistic safety assessment of nuclear power plants. *Annals of Nuclear Energy, 52*, 113-124. doi:DOI 10.1016/j.anucene.2012.08.001

Andrieu, C., & Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing, 18*(4), 343-373. doi:DOI 10.1007/s11222-008-9110-y

Bier, V. M., Haimes, Y. Y., Lambert, J. H., Matalas, N. C., & Zimmerman, R. (1999). A survey of approaches for assessing and managing the risk of extremes. *Risk Analysis, 19*(1), 83-94.

Catalyurek, U., Rutt, B., Metzroth, K., Hakobyan, A., Aldemir, T., Denning, R., . . . Kunsman, D. (2010). Development of a code-agnostic computational infrastructure for the dynamic generation of accident progression event trees. *Reliability Engineering & System Safety, 95*(3), 278-294. doi:DOI 10.1016/j.ress.2009.10.008

Čepin, M., & Mavko, B. (2002). A dynamic fault tree. *Reliability Engineering and System Safety, 75*(1), 83-91. doi:10.1016/S0951-8320(01)00121-1

Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm. *American Statistician, 49*(4), 327-335. doi:Doi 10.2307/2684568

Cojazzi, G. (1996). The DYLAM approach for the dynamic reliability analysis of systems. *Reliability Engineering & System Safety, 52*(3), 279-296.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed.). Hoboken, N.J.: Wiley-Interscience.

Devooght, J., & Smidts, C. (1992). Probabilistic Reactor Dynamics .1. The Theory of Continuous Event Trees. *Nuclear Science and Engineering, 111*(3), 229-240.

Di Maio, F., Baronchelli, S., & Zio, E. (2015a). A Computational Framework for Prime Implicants Identification in Noncoherent Dynamic Systems. *Risk Analysis, 35*(1), 142-156. doi:Doi 10.1111/Risa.12251

Di Maio, F., Vagnoli, M., & Zio, E. (2015). Risk-Based Clustering for Near Misses Identification in Integrated Deterministic and Probabilistic Safety Analysis. *Science and Technology of Nuclear Installations, 501*, 693891.

Fong, C. J., Apostolakis, G. E., Langewisch, D. R., Hejzlar, P., Todreas, N. E., & Driscoll, M. J. (2009). Reliability analysis of a passive cooling system using a response surface with an application to the flexible conversion ratio reactor. *Nuclear Engineering and Design, 239*(12), 2660-2671. doi:DOI 10.1016/j.nucengdes.2009.07.008

Gao, L. J., Liu, S. Y., & Dougal, R. A. (2002). Dynamic lithium-ion battery model for system simulation. *Ieee Transactions on Components and Packaging Technologies, 25*(3), 495-505. doi:Doi 10.1109/Tcapt.2002.803653

Garrett, C., & Apostolakis, G. (1999). Context in the risk assessment of digital systems. *Risk Analysis, 19*(1), 23-32.

Hakobyan, A., Aldemir, T., Denning, R., Dunagan, S., Kunsman, D., Rutt, B., & Catalyurek, U. (2008). Dynamic generation of accident progression event trees. *Nuclear Engineering and Design, 238*(12), 3457-3467. doi:DOI 10.1016/j.nucengdes.2008.08.005

Hansen, A. D., Jauch, C., Sarensen, P. E., Iov, F., & Blaabjerg, F. (2004). *Dynamic wind turbine models in power system simulation tool DIgSILENT*. Roskilde.

Hsueh, K. S., & Mosleh, A. (1996). The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants. *Reliability Engineering & System Safety, 52*(3), 297-314. doi:Doi 10.1016/0951-8320(95)00140-9

Hu, Y. (2005). *A guided simulation methodology for dynamic probabilistic risk assessment of complex systems.* (PhD), University of Maryland, College Park, MD.

Hu, Y., Groen, F., & Mosleh, A. (2004). *An entropy-based exploration strategy in dynamic PRA.* Paper presented at the Probabilistic Safety Assessment and Management.

Kloos, M., & Peschke, J. (2006). MCDET: A probabilistic dynamics method combining Monte Carlo simulation with the discrete dynamic event tree approach. *Nuclear Science and Engineering, 153*(2), 137-156.

Labeau, P. E., Smidts, C., & Swaminathan, S. (2000). Dynamic reliability: towards an integrated platform for probabilistic risk assessment. *Reliability Engineering & System Safety, 68*(3), 219-254. doi:Doi 10.1016/S0951-8320(00)00017-X

Li, J. H., Kang, R., Mosleh, A., & Pan, X. (2011). Simulation-based automatic generation of risk scenarios. *Journal of Systems Engineering and Electronics, 22*(3), 437-444. doi:DOI 10.3969/j.issn.1004-4132.2011.03.011

Mackay, D. J. C. (1992). Information-Based Objective Functions for Active Data Selection. *Neural Computation, 4*(4), 590-604. doi:DOI 10.1162/neco.1992.4.4.590

Mercurio, D., Podofillini, L., Zio, E., & Dang, V. (2009). Identification and classification of dynamic event tree scenarios via possibilistic clustering: application to a steam generator tube rupture event. *Accident Analysis & Prevention, 41*(6), 1180-1191.

Pate-Cornell, E. (2002). Finding and fixing systems weaknesses: probabilistic methods and applications of engineering risk analysis. *Risk Analysis, 22*(2), 319-334.

Paté-Cornell, E. (2012). On "Black Swans" and "Perfect Storms": risk analysis and management when statistics are not enough. *Risk Analysis, 32*(11), 1823-1833.

Perez, M., Reventos, F., Batet, L., Guba, A., Toth, I., Mieusset, T., . . . Del Nevo, A. (2011). Uncertainty and sensitivity analysis of a LBLOCA in a PWR Nuclear Power Plant: Results of the Phase V of the BEMUSE programme. *Nuclear Engineering and Design, 241*(10), 4206-4222. doi:DOI 10.1016/j.nucengdes.2011.08.019

RELAP5-3D. (2005). RELAP5-3D. Idaho Falls, Idaho: Idaho National Laboratory.

Robert, C. P., & Casella, G. (2004). *Monte Carlo statistical methods* (2nd ed.). New York: Springer.

Roberts, G. O., & Rosenthal, J. S. (2009). Examples of Adaptive MCMC. *Journal of Computational and Graphical Statistics, 18*(2), 349-367. doi:DOI 10.1198/jcgs.2009.06134

Rutt, B., Catalyurek, U., Hakobyan, A., Metzroth, K., Aldemir, T., Denning, R., . . . Kunsman, D. (2006). Distributed dynamic event tree generation for reliability and risk assessment. *Challenges of Large Applications in Distributed Environments, Proceedings*, 61-70.

Siu, N. (1994). Risk Assessment for Dynamic-Systems - an Overview. *Reliability Engineering & System Safety, 43*(1), 43-73. doi:Doi 10.1016/0951-8320(94)90095-7

Smidts, C., & Devooght, J. (1992). Probabilistic Reactor Dynamics .2. A Monte-Carlo Study of a Fast-Reactor Transient. *Nuclear Science and Engineering, 111*(3), 241-256.

Turati, P., Pedroni, N., & Zio, E. (2015). *An entropy-driven method for exploring extreme and unexpected accident scenario in the risk assessment of dynamic engineered systems.* Paper presented at the Proceedings of the 25th ESREL, Safety and Reliability of Complex Engineered Systems, Zurich, Swiss.

Zio, E. (2014). Integrated deterministic and probabilistic safety assessment: Concepts, challenges, research directions. *Nuclear Engineering and Design, 280*, 413-419. doi:DOI 10.1016/j.nucengdes.2014.09.004