The Road from Stochastic Automata to the Simulation of Rare Events

Pedro R. D'Argenio[1,2,3(✉)], Carlos E. Budde[4], Matias David Lee[1], Raúl E. Monti[1,2], Leonardo Rodríguez[1], and Nicolás Wolovick[1]

[1] Universidad Nacional de Córdoba, Córdoba, Argentina
dargenio@famaf.unc.edu.ar
[2] CONICET, Córdoba, Argentina
[3] Saarland University, Saarbrücken, Germany
[4] University of Twente, Enschede, The Netherlands

Abstract. We report in the advances on stochastic automata and its use on rare event simulation. We review and introduce an extension of IOSA, an input/output variant of stochastic automata that under mild constraints can be ensured to contain non-determinism only in a spurious manner. That is, the model can be regarded as fully probabilistic and hence amenable for simulation. We also report on our latest work on fully automatizing the technique of rare event simulation. Using the structure of the model given in terms a network of IOSAs allows us to automatically derive the importance function, which is crucial for the importance splitting technique of rare event simulation. We conclude with experimental results that show how promising our technique is.

1 Introduction

Stochastic automata were introduced by D'Argenio et al. in [10] as the semantics basis for the compositional modeling of stochastically timed systems where the occurrence time of events responds to continuous distributions. They can be seen as a variant of timed automata [1] where clocks are initialized randomly and run backwards, enabling transitions as soon as their value become 0. Based on LOTOS [2] and other process algebras, the first ideas for compositionality for stochastic automata were introduced through the process algebra ♤ . Thus, stochastic automata and ♤ provide a natural generalization of generalized semi-Markov processes (GSMP) oriented to compositional modeling.

However, this framework came with the usually unavoidable non-determinism introduced by concurrency. This is a drawback, since, when deterministic, this type of general models could be only analyzed through discrete event simulation for the big majority of quantitative or even qualitative properties. (Model checking stochastic automata can only provide a rough over approximation and even though, with the usual limitation given by the state space explosion [19].) Unfortunately, simulation and non-determinism are incompatible since simulation requires that all possible execution choices are resolve through randomization. This is partly solved in stochastic automata by the races on random clocks

J.-P. Katoen et al. (Eds.): Brinksma Festschrift, LNCS 10500, pp. 276–294, 2017.
DOI: 10.1007/978-3-319-68270-9_14

enabling the transitions. Yet situations like the same clock enabling two different transitions may happen which yields a non-deterministic choice. Notwithstanding this situation, [12] presented a first approach to the simulation of stochastic automata where a scheduler indicating how the non-determinism should be resolved is explicitly required as input.

Notice however that the scheduler is an artifact that becomes part of the model and should be provided by an expert that understands the intricacies of the model. This task is clearly prone to error. Therefore, we sought instead for a way to ensure that the model is fully probabilistic (or deterministic, meaning here that all choices are resolved randomly) by construction. In [13] we introduced input/output stochastic automata (IOSA), a variant of stochastic automata that splits actions into inputs and outputs and let them behave in a reactive and generative manner respectively (see [18] for the concepts of reactive and generative transitions), following ideas proposed in [33]. Since outputs behave generatively, we let their occurrence time be controlled by a random variable (encoded in a clock). As inputs are reactive, they are passive and hence their occurrence time can only depend on their interaction with outputs. Thus, IOSA combines in a single model the two interpretations of stochastic automata (either as open or as closed systems [8,9].) It turns out that after all components are synchronized and the system is closed (i.e. all interactions are resolved), the whole model becomes fully probabilistic (i.e., it does not contain non-determinism).

This variant, however, turned to be a little too restricting for modeling. Decoupling stochastic behavior and synchronization as in [20] may simplify considerably compositional modeling. Thus, in this paper we extend IOSA by allowing certain non-determinism so that we can easily check whether it is spurious, that is, any possible path on the non-deterministic choice will converge to the same state without changing the value of the property. We do this by including urgent or committed transitions that do not take time, allowing that they are non-deterministic, but requesting that they are also confluent (with the standard notion of confluence in concurrency theory [24]). Having obtained a deterministic model, we are in conditions to simulate a closed IOSA with committed actions.

Since, nowadays, systems are required to have a high degree of resilience and dependability, determining properties that fail with extremely small probability in complex models can be computationally very demanding. However, standard Monte Carlo simulation is impractical when the probability of the event under analysis is extremely low: it will easily require an enormous amount of sampling to obtain an acceptable confidence level of the estimated probability, in order to compensate for the high variance induced by the rare occurrences of such event.

To reduce this considerable need for simulation runs, efficient Monte Carlo simulation techniques have been tailored to deal with rare events. These can be largely divided into two conceptually different techniques: *importance sampling* and *importance splitting* methods. We focus on *importance splitting* techniques, see e.g. [23,29,30]. Importance splitting works by decomposing the state space in multiple levels where, ideally, the rare event is at the top level and the probability of (reaching) the rare event increases for each increasing level. The estimation

of the rare probability is obtained as the product of the estimates of the (not so rare) conditional probabilities of moving one level up. As a consequence, the effectiveness of this technique crucially depends on an adequate grouping of states into levels. *Importance functions* are the means to assign a value to each state so that, if perfect, such value is directly related to the likelihood of reaching the rare event. It is desirable that a state in the rare set receives the highest importance and the importance of a state decreases according to the probability of reaching a rare state from it. Usually, an expert in the area of the system provides the importance function in an *ad-hoc* manner. A badly chosen function can deteriorate the effectiveness of the technique. With some notable exceptions [4,16,21,25], automatic derivation of importance functions has received scarce attention.

In the same way that we eliminate the need for an expertise in the modeling of a scheduler, we have looked for techniques to automatically derive such importance functions. The overall aim thus is that the task of rare event simulation becomes a single push button technique after the modeling of the system and the property under study. In [4] we presented preliminary results on an effective technique to derive automatically an importance function. The algorithm works by applying inverse breadth first search (BFS) on the underlying graph of the stochastic process, labeling each state with the shortest distance to a rare state. The importance of each state is then defined as the difference between the maximum distance and its actual distance. This technique still requires a finite graph which fits in the computer memory. Unfortunately such graph grows exponentially with the number of components in the model of the system. To overcome this problem, in [5] we improve on this technique by obtaining the importance function in a compositional manner. We consider the system modeled as a network of IOSAs. The technique then works by applying the previous method per component, previous analysis on how the local states relate to the property under study, and the final importance function is obtained by composing the modular functions. Contrarily to the first technique, this way of calculating the importance function grows linearly with the number of modules that conform the system model. In this paper, we also report on these techniques and show experimental studies that demonstrate how promising our ideas are.

2 Input/Output Stochastic Automata

Stochastic automata [8–10] use continuous random variables called clocks to observe the passage of time and control the occurrence of events. This variables are set to a value according to their associated probability distribution, and as time evolves, they count down at the same rate. When a clock reaches zero, it may trigger some action. This allows the modeling of systems where events occur at random continuous time stamps.

Following ideas from [33], input/output stochastic automata (IOSA for short) restrict stochastic automata by splitting actions into input and output actions which will act in a reactive and generative way respectively [18]. This splitting

reflects the fact that input actions are considered to be controlled externally, while output actions are locally controlled. Therefore, we consider the system input enabled. Moreover, output actions could be stochastically controlled or instantaneous. In the first case output actions are controlled by the expiration of a single clock while in the second case the output actions take place as soon as the enabling state is reached. We called these instantaneous actions *committed*. A set of restrictions over IOSA will ensure that, almost surely, no two non committed outputs are enabled at the same time.

Definition 1. *An input/output stochastic automaton with committed actions (IOSA for short) is a structure* $(\mathcal{S}, \mathcal{A}, \mathcal{C}, \rightarrow, C_0, s_0)$*, where* $\mathcal{S}$ *is a (denumerable) set of states,* $\mathcal{A}$ *is a (denumerable) set of labels partitioned into disjoint sets of input labels* $\mathcal{A}^I$ *and output labels* $\mathcal{A}^O$*, from which a subset* $\mathcal{A}^{co}$ *of them are marked as committed,* $\mathcal{C}$ *is a (finite) set of clocks where each* $x \in \mathcal{C}$ *has associated a continuous probability measure* μ_x *on* $\mathbb{R}$ *s.t.* $\mu_x(\mathbb{R}_{>0}) = 1$*,* $\rightarrow \subseteq \mathcal{S} \times 2^{\mathcal{C}} \times \mathcal{A} \times 2^{\mathcal{C}} \times S$ *is a transition function,* C_0 *is the set of clocks that are initialized in the initial state, and* $s_0 \in \mathcal{S}$ *is the initial state.*

In addition, a IOSA should satisfy the following constraints, where we write $s \xrightarrow{C,a,C'} s'$ *instead of* $(s, C, a, C', s') \in \rightarrow$*:*

(a) *If* $s \xrightarrow{C,a,C'} s'$ *and* $a \in \mathcal{A}^I \cup \mathcal{A}^{co}$*, then* $C = \emptyset$*.*

(b) *If* $s \xrightarrow{C,a,C'} s'$ *and* $a \in \mathcal{A}^O \setminus \mathcal{A}^{co}$*, then* C *is a singleton set.*

(c) *If* $s \xrightarrow{\{x\},a_1,C_1} s_1$ *and* $s \xrightarrow{\{x\},a_2,C_2} s_2$ *then* $a_1 = a_2$*,* $C_1 = C_2$ *and* $s_1 = s_2$*.*

(d) *If* $s \xrightarrow{\{x\},a,C} s'$ *then* $x \in \bigcup \text{safe}(s)$*, where* safe *is the least fixed point of* F *defined as:*

$$\mathbf{F}(\mathsf{X})(s) = \{C_0 \mid s = s_0\} \cup \{C' \cup (\{y \mid \hat{s} \xrightarrow{\{y\},\_,\_} \_\} \setminus C) \mid \hat{s} \xrightarrow{C,a,C'} s \wedge a \notin \mathcal{A}^{co}\}$$
$$\cup \{C \cup C' \mid \hat{s} \xrightarrow{\emptyset,a,C} s \wedge a \in \mathcal{A}^{co} \wedge C' \in \mathsf{X}(\hat{s})\}$$

(e) *For every* $a \in \mathcal{A}^I$ *and state* s*, there exists a transition* $s \xrightarrow{\emptyset,a,C} s'$*.*

(f) *For every* $a \in \mathcal{A}^I$*, if* $s \xrightarrow{\emptyset,a,C'_1} s_1$ *and* $s \xrightarrow{\emptyset,a,C'_2} s_2$*,* $C'_1 = C'_2$ *and* $s_1 = s_2$*.*

The occurrence of a transition is controlled by the expiration of clocks. $s \xrightarrow{C,a,C'} s'$ indicates that there is a transition from state s to state s' that can be taken only when all clocks in C have expired and, when taken, it triggers action a and sets all clocks in C' to a value sampled from their associated probability distribution. We write _ to replace parameters when they are not relevant.

These restrictions ensure that any *closed* IOSA without committed actions is deterministic [13]. An IOSA is closed if all its synchronizations have been resolved, that is, the IOSA resulting from a composition does not have input actions ($\mathcal{A}^I = \emptyset$).

Restriction (a) is two-folded: on the one hand, it specifies that input actions are reactive and their time occurrence can only depend on the interaction with an output, on the other hand, committed output actions must occur as soon as

the state enables them. The difference will be more clear when we define the concrete semantics. Restriction (b) specifies that each non-committed output is locally controlled and has a single associated clock which controls its occurrence. Restriction (c) ensures that different non-committed output actions leaving the same state cannot be controlled by the same clock. Restriction (e) ensures input enabling. Restriction (f) determines that IOSAs are input deterministic. Therefore, the same input action in the same state can not jump to different states, nor set different clocks.

Finally, restriction (d) restricts enabling clock x to clocks that have not yet expired when reaching s. That is, either x has been reset during the transition to s, or during a path of committed transitions reaching s, or x is not used as enabling clock of a transition to s but it is an enabling clock on the immediately preceding state. By means of the least fixed point of $\mathbf{F}$ we are able to accumulate clocks that are reset along paths of committed transitions. Furthermore, this restriction allows a clock x to be an enabling clock at an initial state s if x is an initial clock, i.e. $x \in C_0$.

Note that since clocks are set by sampling from a continuous random variables, the probability that the values of two different clocks are equal is 0. This fact along with restriction (c) and (d) guarantees that almost never two different non-committed output transitions are enabled at the same time.

In the following we define parallel composition of IOSAs. Since we intend outputs to be autonomous (or locally controlled), we do not allow synchronization between them. Besides, we need to avoid name clashes on the clocks, so that the intended behavior of each component is preserved and moreover, to ensure that the resulting composed automaton is indeed an IOSA. Furthermore, synchronizing IOSAs should agree on committed actions in order to ensure their immediate occurrence. Thus we require to compose only *compatible* IOSAs.

Definition 2. *Two IOSAs $\mathcal{I}_1$ and $\mathcal{I}_2$ are said to be compatible if they do not share output actions nor clocks, i.e. $\mathcal{A}_1^O \cap \mathcal{A}_2^O = \emptyset$ and $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$, and moreover they agree on committed actions, i.e. $\mathcal{A}_1 \cap \mathcal{A}_2^{co} = \mathcal{A}_2 \cap \mathcal{A}_1^{co}$.*

Definition 3. *Given two compatible IOSAs $\mathcal{I}_1$ and $\mathcal{I}_2$, the parallel composition $\mathcal{I}_1||\mathcal{I}_2$ is a new IOSA $(\mathcal{S}_1 \times \mathcal{S}_2, \mathcal{A}, \mathcal{C}, \rightarrow, C_0, s_0^1||s_0^2)$ where (i) $\mathcal{A}^O = \mathcal{A}_1^O \cup \mathcal{A}_2^O$, (ii) $\mathcal{A}^I = (\mathcal{A}_1^I \cup \mathcal{A}_2^I) \setminus \mathcal{A}^O$, (iii) $\mathcal{A}^{co} = \mathcal{A}_1^{co} \cup \mathcal{A}_2^{co}$, (iv) $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$, and (v) $C_0 = C_0^1 \cup C_0^2$, and $\rightarrow$ is the smallest relation defined by rules in Table 1 where we annotate $s||t$ instead of (s,t).*

Table 1. Parallel composition on IOSA

$$\frac{s_1 \xrightarrow{C,a,C'}_1 s_1'}{s_1||s_2 \xrightarrow{C,a,C'} s_1'||s_2}\ a \in \mathcal{A}_1 \setminus \mathcal{A}_2 \qquad (1) \qquad\qquad \frac{s_2 \xrightarrow{C,a,C'}_2 s_2'}{s_1||s_2 \xrightarrow{C,a,C'} s_1||s_2'}\ a \in \mathcal{A}_2 \setminus \mathcal{A}_1 \qquad (2)$$

$$\frac{s_1 \xrightarrow{C_1,a,C_1'}_1 s_1' \quad s_2 \xrightarrow{C_2,a,C_2'}_2 s_2'}{s_1||s_2 \xrightarrow{C_1 \cup C_2,a,C_1' \cup C_2'} s_1'||s_2'} \qquad (3)$$

It can be proven that the parallel composition preserves IOSAs. That is, the parallel composition of two IOSAs is also an IOSA.

Following ideas from Milner [24] we say that an IOSA is confluent with respect to actions a and b if the occurrence of one of them does not prevent the other one from occurring in the future. More precisely, an IOSA $\mathcal{I}$ is confluent with respect to committed actions a and b in $\mathcal{A}$ if for every state s in $\mathcal{S}$ we can complete the diagram from Fig. 1.

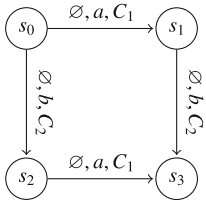

Fig. 1. Confluence in IOSA.

Notice that confluent actions do not alter the stochastic behavior of the system: by considering a and b silent actions (i.e. $a = b = \tau$ with τ interpreted as in Milner's work [24]) the IOSA of Fig. 1 behaves like the single transition $s_0 \xrightarrow{\emptyset,\tau,C_1 \cup C_2} s_3$. Thus, the non-determinism introduced by confluent committed actions is spurious.

It can be shown that parallel composition preserves confluence. Thus, if all IOSA components are confluent for all committed action, so is their parallel composition.

3 Semantics of IOSA

The semantics of IOSA is defined in terms of non-deterministic labeled Markov processes (NLMP) [14,32]. A NLMP is a generalization of probabilistic transition systems with continuous domain. In particular, it extends LMP [15] with *internal* non-determinism.

The foundations of NLMP is strongly rooted in measure theory, hence we recall first some basic definitions. Given a set S and a collection Σ of subsets of S, we call Σ a *σ-algebra* iff $S \in \Sigma$ and Σ is closed under complement and denumerable union. We call the pair (S, Σ) a *measurable space*. A function $\mu : \Sigma \to [0, 1]$ is a *probability measure* if (i) $\mu(\bigcup_{i \in \mathbb{N}} Q_i) = \sum_{i \in \mathbb{N}} \mu(Q_i)$ for all countable family of pairwise disjoint measurable sets $\{Q_i\}_{i \in \mathbb{N}} \subseteq \Sigma$, and (ii) $\mu(S) = 1$. In particular, for $s \in S$, δ_s denotes the Dirac measure so that $\delta_s(\{s\}) = 1$. Let $\Delta(S)$ denote the set of all probability measures over (S, Σ). Let (S_1, Σ_1) and (S_2, Σ_2) be two measurable spaces. A function $f : S_1 \to S_2$ is said to be *measurable* if for all $Q_2 \in \Sigma_2$, $f^{-1}(Q_2) \in \Sigma_1$. There is a standard construction by Giry [17] to endow $\Delta(S)$ with a σ-algebra as follows: $\Delta(\Sigma)$ is defined as the smallest σ-algebra containing the sets $\Delta^B(Q) \doteq \{\mu \mid \mu(Q) \in B\}$, with $Q \in \Sigma$ and $B \in \mathscr{B}([0, 1])$, where $\mathscr{B}([0, 1])$ is the usual Borel σ-algebra on the interval $[0, 1]$. Finally, we define the *hit σ-algebra* $H(\Delta(\Sigma))$ as the minimal σ-algebra containing all sets $H_\xi = \{\zeta \in \Delta(\Sigma) \mid \zeta \cap \xi \neq \emptyset\}$ with $\xi \in \Delta(\Sigma)$.

Definition 4. *A* non-deterministic labeled Markov process *(NLMP for short) is a structure* $(\boldsymbol{S}, \Sigma, \{\mathcal{T}_a \mid a \in \mathcal{L}\})$ *where* Σ *is a σ-algebra on the set of states* $\boldsymbol{S}$*, and for each label* $a \in \mathcal{L}$ *we have that* $\mathcal{T}_a : \boldsymbol{S} \to \Delta(\Sigma)$ *is measurable from* Σ *to* $H(\Delta(\Sigma))$.

The formal semantics of an IOSA is defined by a NLMP with two classes of transitions: one that encodes the discrete steps and contains all the probabilistic information introduced by the sampling of clocks, and another describing the time steps, that only records the passage of time synchronously decreasing the value of all clocks. For simplicity, we assume that the set of clocks has a particular order and their current values follow the same order in a vector.

Definition 5. *Given an IOSA* $\mathcal{I} = (\mathcal{S}, \mathcal{A}, \mathcal{C}, \rightarrow, C_0, s_0)$ *with* $\mathcal{C} = \{x_1, \ldots, x_N\}$, *its semantics is defined by the NLMP* $\mathcal{P}(\mathcal{I}) = (\boldsymbol{S}, \mathscr{B}(\boldsymbol{S}), \{\mathcal{T}_a \mid a \in \mathcal{L}\})$ *where*

- $\boldsymbol{S} = (\mathcal{S} \cup \{\mathsf{init}\}) \times \mathbb{R}^N$, $\mathcal{L} = \mathcal{A} \cup \mathbb{R}_{>0} \cup \{\mathsf{init}\}$, *with* $\mathsf{init} \notin \mathcal{S} \cup \mathcal{A} \cup \mathbb{R}_{>0}$
- $\mathcal{T}_{\mathsf{init}}(\mathsf{init}, \boldsymbol{v}) = \{\delta_{s_0} \times \prod_{i=1}^{N} \mu_{x_i}\}$,
- $\mathcal{T}_a(s, \boldsymbol{v}) = \{\mu_{\boldsymbol{v},C',s'} \mid s \xrightarrow{C,a,C'} s', \bigwedge_{x_i \in C} \boldsymbol{v}(i) \leq 0\}$, *for all* $a \in \mathcal{A}$, *where* $\mu_{\boldsymbol{v},C',s'} = \delta_{s'} \times \prod_{i=1}^{N} \overline{\mu}_{x_i}$ *with* $\overline{\mu}_{x_i} = \mu_{x_i}$ *if* $x_i \in C'$ *and* $\overline{\mu}_{x_i} = \delta_{\boldsymbol{v}(i)}$ *otherwise, and*
- $\mathcal{T}_d(s, \boldsymbol{v}) = \{\delta_s \times \prod_{i=1}^{N} \delta_{\boldsymbol{v}(i)-d}\}$ *if* $s \overset{b}{\not\rightarrow}$ *for all committed* $b \in \mathcal{A}^O \cap \mathcal{A}^{co}$ *and* $0 < d \leq \min\{\boldsymbol{v}(i) \mid \exists a \in \mathcal{A}^O, C' \subseteq \mathcal{C}, s' \in S : s \xrightarrow{\{x_i\},a,C'} s'\}$, *and* $\mathcal{T}_d(s, \boldsymbol{v}) = \emptyset$ *otherwise, for all* $d \in \mathbb{R}_{\geq 0}$.

The state space is the product space of the states of the IOSA with all possible clock valuations. A distinguished initial state init is added to encode the random initialization of all clocks (it would be sufficient to initialize clocks in C_0 but we decided for this simplification). Such encoding is done by transition $\mathcal{T}_{\mathsf{init}}$. The state space is structured with the usual Borel σ-algebra. The discrete step is encoded by $\mathcal{T}_a$, with $a \in \mathcal{A}$. Notice that, at state $(s, \boldsymbol{v})$, the transition $s \xrightarrow{C,a,C'} s'$ will only take place if $\bigwedge_{x_i \in C} \boldsymbol{v}(i) \leq 0$, that is, if the current values of all clocks in C are not positive. For the particular case of the input or committed actions this will always be true. The next actual state would be determined randomly as follows: the symbolic state will be s' (this corresponds to $\delta_{s'}$ in $\mu_{\boldsymbol{v},C',s'} = \delta_{s'} \times \prod_{i=1}^{N} \overline{\mu}_{x_i}$), any clock not in C' preserves the current value (hence $\overline{\mu}_{x_i} = \delta_{\boldsymbol{v}(i)}$ if $x_i \notin C'$), and any clock in C' is set randomly according to its respective associated distribution (hence $\overline{\mu}_{x_i} = \mu_{x_i}$ if $x_i \in C'$). The time step is encoded by $\mathcal{T}_d(s, \boldsymbol{v})$ with $d \in \mathbb{R}_{\geq 0}$. It can only take place at d units of time if there is no output transition enabled at the current state within the next d time units (this is verified by condition $0 < d \leq \min\{\boldsymbol{v}(i) \mid \exists a \in \mathcal{A}^O, C' \subseteq \mathcal{C}, s' \in S : s \xrightarrow{\{x_i\},a,C'} s'\}$). In this case, the system remains in the same symbolic state (this corresponds to δ_s in $\delta^{-d}_{(s,\boldsymbol{v})} = \delta_s \times \prod_{i=1}^{N} \delta_{\boldsymbol{v}(i)-d}$), and all clock values are decreased by d units of time (represented by $\delta_{\boldsymbol{v}(i)-d}$ in the same formula). Note the difference from the timed transitions semantics of pure IOSA [13]. This is due to the maximal progress assumption, which forces to take committed transition as soon as they get enabled. We encode this by not allowing to make time transitions in presence of committed actions, i.e. $s \overset{b}{\not\rightarrow}$ for all committed $b \in \mathcal{A}^O \cap \mathcal{A}^{co}$ (thus $\mathcal{T}_d(s, \boldsymbol{v}) = \emptyset$ whenever $s \xrightarrow{C,b,C'} s'$ with $b \in \mathcal{A}^O \cap \mathcal{A}^{co}$.) Instead, notice the *patient* nature of a state $(s, \boldsymbol{v})$ that has no output enabled.

That is, $\mathcal{T}_d(s, \boldsymbol{v}) = \{\delta_s \times \prod_{i=1}^{N} \delta_{\boldsymbol{v}(i)-d}\}$ for all $d > 0$ whenever $s \overset{b}{\not\rightarrow}$ for all output action $b \in \mathcal{A}^O$.

In a similar way to [13], it is possible to show that $\mathcal{P}(\mathcal{I})$ is indeed a NLMP, i.e. that $\mathcal{T}_a$ maps into measurable sets in $\Delta(\mathscr{B}(\mathbf{S}))$, and that $\mathcal{T}_a$ is a measurable function for every $a \in \mathcal{L}$.

4 Rare Event Simulation

Assuming that the IOSA is closed and confluent on all committed actions and it does not contain loops of only committed transitions, from the semantics of IOSA (Definition 5) we can extract an algorithm for discrete event simulation which we give in Fig. 2.

Given that the IOSA is confluent for committed actions, the arbitrary choice of a committed transition in line 6 is irrelevant since, after finishing the while loop of line 5, the same set of clocks will be sampled whichever path of committed transitions is taken. Moreover, the while loop is ensured to finish since no loop of committed transition is allowed. Also, the restrictions imposed by Definition 1 guarantee the uniqueness of the transition in line 11 [13].

When a parameter is estimated using the usual Monte Carlo simulation (as described in Fig. 2), the speed and overall efficiency of the method is highly dependent on the precision required for the estimate. Confidence intervals are commonly used to convey a notion of how far the produced estimate may be from the actual value. As a general rule, whichever the confidence interval construction method, the simulations "length" grows with the tightness desired for

1: **for** all clock $x \in C_0$ **do**
2: Sample a value $v(x)$ according to the distribution μ_x
3: Set s_0 as current state s.
4: **repeat**
5: **while** s has an outgoing committed transition **do**
6: Choose any committed transition $s \xrightarrow{\varnothing,a,C} s'$.
7: **for** all clock $x \in C_0$ **do**
8: Sample a value $v(x)$ according to the distribution μ_x
9: Set s' as current state s.
10: Let $x \in C$ be the clock with the smallest positive value.
11: Let $s \xrightarrow{\{x\},a,C} s'$ be the unique transition enabled by x.
12: **for** all clock $y \notin C$ **do**
13: $v(y) = v(y) - v(x)$
14: **for** all clock $y \in C$ **do**
15: Sample a value $v(y)$ according to the distribution μ_y
16: Set s' as current state s.
17: Collect the relevant statistical data.
18: **until** sufficient statistical data was collected

Fig. 2. Simulation of a closed confluent IOSA

the interval. In particular several rare event scenarios are known to require a number of samples which grows exponentially on the model size [22].

Importance splitting (IS for short) aims to speed up the occurrence of a rare event without modifications on the system dynamics (see [23] and references therein.) The general idea in IS is to favor the "promising runs" that approach the rare event by saving the states they visit at certain predefined checkpoints. Replicas of these runs are created from those checkpoint states, which continue evolving independently from then on. Contrarily, simulation runs deemed to steer away from the rare event are identified and killed, avoiding the use of computational power in fruitless calculi. The likelihood of visiting a goal state from any other state s is called the *importance* of s. Variations in such importance determine when should a simulation run be split or killed, as the importance value crosses some given *thresholds* up or down, respectively.

We focus on the RESTART method, a version of IS with multiple thresholds, fixed splitting and deterministic discards of unpromising simulations [26,28–31]. A RESTART run can be depicted as in Fig. 3 where the horizontal axis represents the simulation progress and the vertical axis the importance value of the current state. The run starts from an initial state and evolves until the first threshold T_1 is crossed *upwards*. This takes the path from zone Z_0 below threshold T_1 into zone Z_1 between T_1 and T_2. As this happens the state is saved and $s_1 - 1$ replicas or *offsprings* of the path are created. See A in Fig. 3, where the *splitting* for T_1 is $s_1 = 3$. This follows the idea of rewarding promising simulations: up-crossing a threshold suggests the path heads towards a goal state. From then on the s_1 simulations will evolve independently. As they continue, one of them may hit the upper threshold T_2, activating the same procedure: $s_2 - 1$ offsprings are generated from it and set to evolve independently. See B in Fig. 3; here, the splitting is $s_2 = 2$.

However, it could also happen that some simulation hits T_1 again, meaning the path is leading *downwards*. This simulation steers away from the goal set and RESTART deals with it discarding the run right away (see C in Fig. 3). In each zone Z_i there exists nonetheless an *original simulation*, which crossed threshold T_i upwards generating the $s_i - 1$ offsprings. This run is allowed to survive a down-crossing of threshold T_i (see D in Fig. 3).

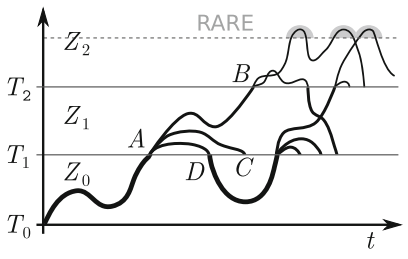

Fig. 3. RESTART importance splitting

In this setting all simulations reaching a goal state went through the replication procedure, which stacked up on every threshold crossed. Simply counting these hits would introduce a bias, because the *relative weight* of the runs in upper zones decreases by an amount equivalent to the splitting of the thresholds. In consequence, each rare event observed is pondered by the relative weight of the simulation from which it stemmed. If all the goal states exist beyond the

uppermost threshold like in Fig. 3, then it suffices to divide the observed quantity of rare events by $\text{SPLIT}_{\mathsf{MAX}} \doteq \prod_{i=1}^{n} s_i$. Otherwise more involved labeling mechanisms are needed.

In this work we study transient and long run properties. *Transient* properties are used to calculate the probability of reaching a set G of *goal states* before visiting any *reset state* from the (disjoint) set R. (For simulation purposes the probability of reaching a state in $G \uplus R$ has to be 1.) Following PCTL, we denote this probability by $\mathsf{P}(\neg R \,\mathsf{U}\, G)$. *Long run* analysis focuses on the quantification of a property once the system has reached an equilibrium. In particular, the steady state probability of a set G of *goal states* is the portion of time in which any state in G is visited in the long run. Using CSL notation, we write $\mathsf{S}(G)$.

5 Automatic Derivation of the Importance Function

Notice that a simulation using importance splitting is entirely guided by the *importance function* which defines the importance of each state. This function conveys the states where the simulation effort should be intensified. Importance functions are defined in most situations in an *ad-hoc* fashion by an expert in the field of the particular system model. With a few exceptions in some specific areas [16,21,25,35] automatic derivation of importance functions is still a novel field for general systems and this has been our later concern [3–5].

Consider a single IOSA and any of the properties $\mathsf{P}(\neg R \,\mathsf{U}\, G)$ or $\mathsf{S}(G)$. The rare event is precisely the set G of *goal states*. In [4], we propose a distance based on the length of the shortest path on the IOSA leading to a state in G: a state s is more important than other state s' if its shortest path to a state in G is shorter than the shortest path of s'. This can be implemented with the help of a breadth-first search algorithm that follows the backwards direction of the transitions in the given IOSA. The algorithm, which is given in Fig. 4, has complexity $\mathcal{O}(n \cdot k)$, where n is the size of the state space and k is the branching degree of the underlying graph of the IOSA.

Input: a IOSA model and
the goal state set $G \neq \varnothing$
$g(G) \leftarrow 0$
`queue.push`(G)
repeat
  $s \leftarrow$ `queue.pop`()
  for all $s' \xrightarrow{C,a,C'} s$ **do**
    if s' not visited **then**
      $g(s') \leftarrow g(s) + 1$
      `queue.push`(s')
until `queue.is_empty`() or s_0 visited
$g(s) \leftarrow g(s_0)$ for every non visited state s
$f(s) \leftarrow g(s_0) - g(s)$ for every state s
return importance function f

Fig. 4. Basic importance function derivation

Using this strategy one can indeed obtain in very short computational time a good importance function to use with the IS technique of choice [4]. The thresholds can then be selected either arbitrarily, using e.g. some fixed approach ("*set one every three importance values*"), or adaptively by means of an algorithm that exercises the model dynamics [6,7].

However, this approach does not scale. The BFS algorithm requires an explicit representation of the state space of the composed IOSA (and actually of the whole adjacency matrix), which grows exponentially with the number of modules involved in the composition. This is clearly not in the spirit of simulation which scales nicely since it only requires to save only the current state been explored.

Taking advantage of the compositional nature of IOSA, in [5] we presented a compositional approach to automatically produce importance functions. The solution reuses our previous idea:

(i) identify the set G_i of local states in each IOSA component $\mathcal{I}_i$ that are part of the global set G of goal states,
(ii) apply the algorithm of Fig. 4 in each component $\mathcal{I}_i$ to obtain a *local importance function* f_i, and
(iii) compose the family of functions $\{f_i\}_i$ to obtain the *(global) importance function* f.

This brings two challenges: obtaining the local goal states sets G_i and composing the family of functions $\{f_i\}_i$ to obtain the importance function f.

For the first challenge, we require that the set G of goal states is given in terms of a propositional formula in *disjunctive normal form* (*DNF*), i.e. a disjunction of *clauses*, each of which is a conjunction of *literals* (i.e. of atomic propositions or negated atomic propositions). As a restriction, we impose that each atomic proposition can only be changed or tested in a single IOSA component. This approach imposes no restriction on the description of the rare event, since any propositional formula can be equivalently written in DNF.

To obtain the set G_i of local goal states for component $\mathcal{I}_i$, we "project" the DNF formula $\bigvee_{n\in N}\bigwedge_{m\in M_n}\ell_{nm}$ defining G as follows. For each $n \in N$ define $L_n = \{\ell_{nm} \mid m \in M_n \text{ and } \ell_{nm} \text{ contains a proposition in } \mathcal{I}_i\}$. Then, define the local goal DNF formula by $\bigvee\{\bigwedge L_n \mid n \in N \text{ and } L_n \neq \emptyset\}$ which defines the set G_i of states of $\mathcal{I}_i$ in which such a formula is valid.

For composing the family of functions $\{f_i\}_i$ into the importance function f, we have experimented with several proposals. One option is to let the user settle the matter via an *ad-hoc* choice. He would have to provide an algebraic expression using the local importance which would be used at every step of the simulation to combine the local importance values. For example, consider a system of three IOSAs composed in parallel. If $s|_i$ denotes the projection of the global state s into the local state of component $\mathcal{I}_i$, possible definitions for f are $f(s) = f_1(s|_1) + f_2(s|_2) + f_3(s|_3)$ or $f(s) = \max(0.3f_1(s|_1) + 0.7f_2(s|_2),\ f_3(s|_3))$.

Since we request the properties to be expressed in DNF, we could exploit the structure of the formula to identify specific arithmetical operands or even algebraic structures to associate to each logical operand. We are currently investigating a way to automatically map the disjunctions and conjunctions to their best-match arithmetical counterparts. Our last studies are leading us towards the use of semi-rings such as `(max,+)` and `(+,*)`, which could be thought of as naturally corresponding to the $(\vee, \wedge)$ structure of DNF formulas. For example, consider a system of three IOSAs composed in parallel, where p_i is a propositional formula

in the component $\mathcal{I}_i$. If the goal DNF formula is $(p_1 \wedge p_2) \vee (p_1 \wedge p_3)$, the importance function could be defined by $f(s) = (f_1(s|_1) * f_2(s|_2)) + (f_1(s|_1) * f_3(s|_3))$ or $f(s) = \max((f_1(s|_1) + f_2(s|_2)), (f_1(s|_1) + f_3(s|_3)))$.

As a final remark notice that using the product to combine local importance functions could lead to problems whenever a null importance value is encountered. As a workaround in such cases the functions where updated after construction, replacing every importance value i with 2^i (e.g. the values $0, 1, 2, \ldots$ map into $1, 2, 4, \ldots$) This solved the issue and set the computed importance values further apart, with interesting consequences in the IS simulations.

6 Experimental Results

We have developed the software tool FIG, which implements the compositional approach to multilevel splitting described above. It is written in C++ and is a standalone software. FIG stands for *Finite Improbability Generator* as a homage to Douglas Adam's masterpiece and it is freely available at http://dsg.famaf.unc.edu.ar/fig.

In the following we report several case studies that validate our approach. All experiments were run in a computer with a 12-cores 2.40 GHz Intel Xeon E5-2620v3 processor, and 128 GiB 2133 MHz of available DDR4 RAM. More details of these and other case studies can be found in [3].

Tandem Queue. This system consists of a Jackson tandem network with two sequentially connected queues, where the rates of arrival, first service and second service are respectively $(\lambda, \mu_1, \mu_2) = (3, 2, 6)$, and for which transient and steady-state properties were evaluated.

Notice this tandem queue is Markovian. Therefore, we were able to validate that the results yielded by FIG because the IOSA model agree with those yielded by PRISM for an equivalent model written in the PRISM language. (We remark that the FIG input language is very much alike the PRISM input language.)

For this case study, we have performed transient and steady state analysis. For the transient analysis, the property of interest is $\mathsf{P}(q_2 > 0 \,\mathsf{U}\, q_2 = C)$, i.e. the likelihood of observing a saturated second queue before it becomes empty, which we estimate starting from the state $(q_1, q_2) = (0, 1)$. We tested maximum queue capacities $C \in \{8, 10, 12, 14\}$, for which the values calculated with PRISM are respectively 5.62e−6, 3.14e−7, 1.86e−8, and 1.14e−9. Estimations were set to achieve 90% confidence interval with 20% of relative error. The execution timeout was 2.5 h, which FIG converged for each configuration producing intervals containing the values reported by PRISM.

The average of the wall times measured in three experiments are shown in Fig. 5. Three different importance functions were tested in the importance splitting simulations. The function denoted `amono` was automatically built by FIG using the monolithic approach of [4]. Instead, `acomp` stands for the function built following the compositional strategy, which in this case employed summation as composition operand (i.e., the global function is the summation of the local functions). The third importance function tested with RESTART was one of the

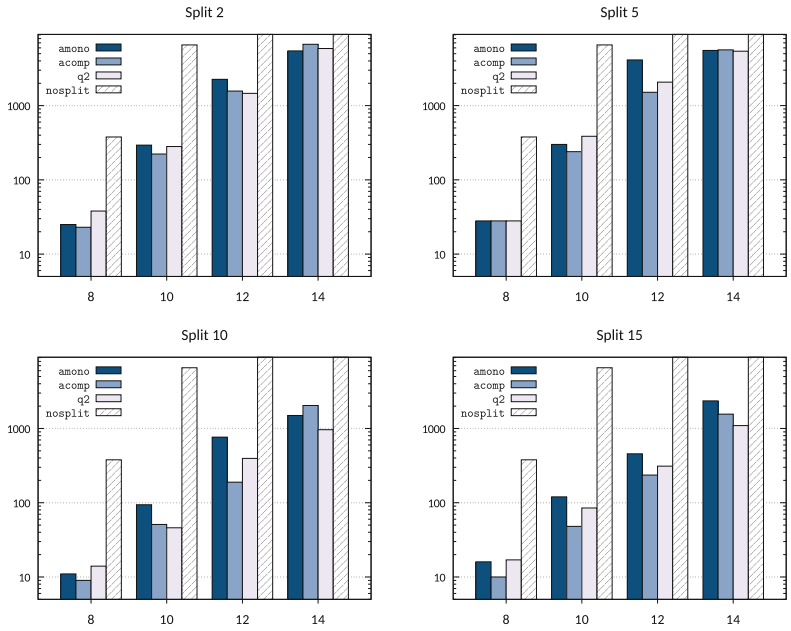

Fig. 5. Times for the transient analysis of the tandem queue

best known *ad-hoc* candidates viz. counting the number of packets in the second queue, which we denote `q2`. Standard Monte Carlo simulations are denoted `nosplit`. In Fig. 5, we display one chart per splitting value, with the outcomes of the `nosplit` simulations repeated in all four charts. The maximum queue capacity C, tuned to variate the rarity of the event, spans along the x-axis.

Regarding long run simulations we are interested in the property $\mathsf{S}(q_2 = C)$, i.e. the proportion of time that the second queue spends in a saturated state. We tested maximum queue capacities $C \in \{10, 13, 16, 18, 21\}$, for which the values calculated with PRISM are respectively 7.25e−6, 2.86e−7, 1.12e−8, 1.28e−9, and 4.94e−11.

Estimations were set to achieve 90% confidence with 20% of relative error and expected to converge within 6 h of wall time execution. Again we corroborated that these estimations converged to the values yielded by PRISM. The same importance functions than in the transient case were employed.

The results obtained from an average among three experiments are presented in Fig. 6, following the same format than in the transient case.

Triple Tandem Queue. Consider a non-Markovian tandem network operating under the same principles than the previous tandem queue, but consisting of *three* queues with *Erlang-distributed* service times. The shape parameter α is the same for all servers, but the scale parameters $\{\mu_i\}_{i=1}^{3}$ differ from one queue to the next. Arrivals into the system are exponential with rate $\lambda = 1$.

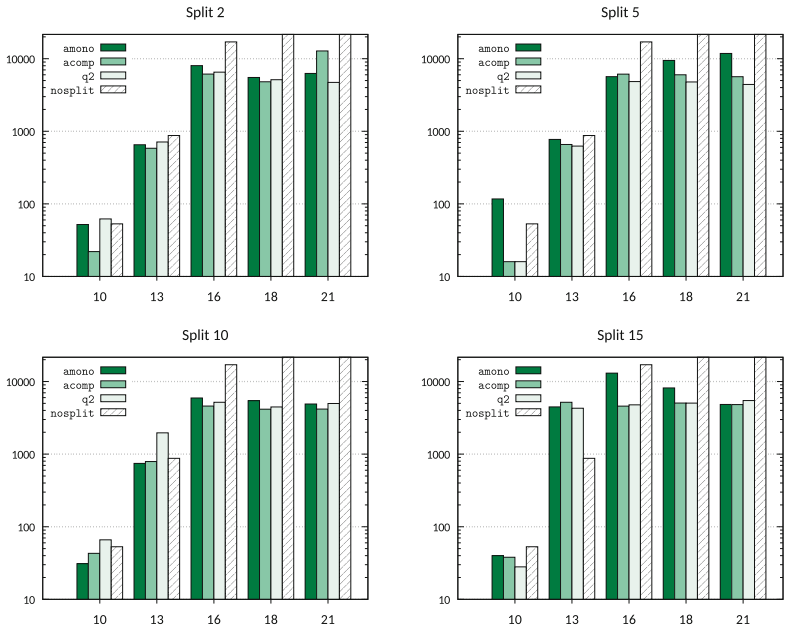

Fig. 6. Times for the steady-state analysis of the tandem queue

The long run behavior of this non-Markovian triple tandem queue was studied in [26] starting from an empty system. The shape parameter is $\alpha \in \{2, 3\}$ in all queues and the load at the third queue is kept at $1/3$. This means that the scale parameter μ_3 in the third queue takes the values $1/6$ and $1/9$ when α is 2 and 3 respectively. The scale parameters μ_1 and μ_2 of the first and second servers, as well as the thresholds capacity C at the third queue, are chosen to keep the steady-state probability in the same order of magnitude for all case studies.

The property of interest is the steady-state probability of a saturation in the third queue, i.e. $\mathsf{S}(q_3 = C)$. Following the same approach from [26] we choose the parameters so that the estimated value is in the order of $5 \cdot 10^{-9}$. Thus the values of $(\alpha, \mu_1, \mu_2, C)$ for the six case studies I–VI are respectively $(2, 1/3, 1/4, 10)$, $(3, 2/3, 1/6, 7)$, $(2, 1/6, 1/4, 11)$, $(3, 1/9, 1/6, 9)$, $(2, 1/10, 1/8, 14)$, and $(3, 1/15, 1/12, 12)$.

Estimations were set to achieve 90% confidence interval with 20% of relative error and expected to converge within 4 h of wall time execution. Four importance functions were tested in the importance splitting simulations: the monolithic (`amono`) and compositional (`acomp`) functions which FIG can build automatically, using summation as composition operand for `acomp`; an *ad-hoc* function which just counted occupation in the third queue (`q3`); and the *ad-hoc* approach from [26] (denoted `jva`), which also considers the occupancy in the other queues with weight coefficients specific to each case, taking values in the interval $[0.2, 0.9]$.

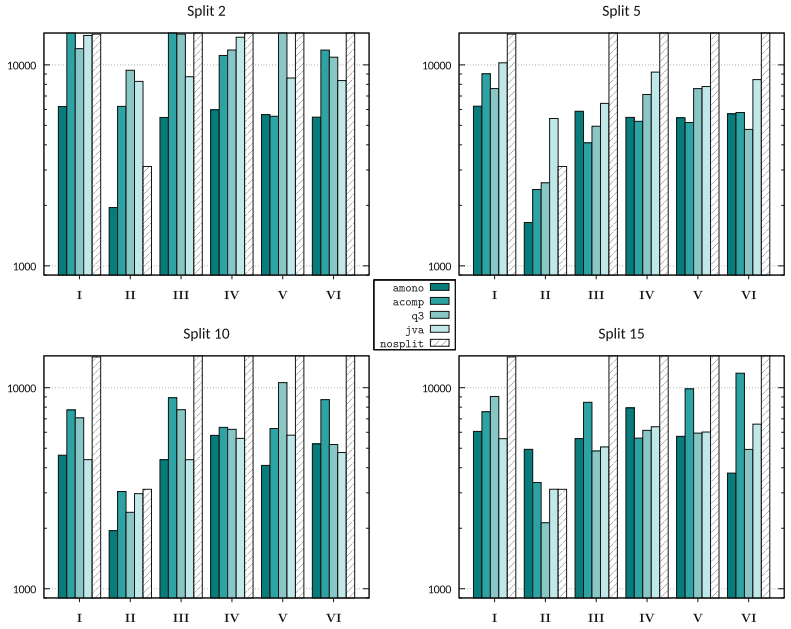

Fig. 7. Times for the steady-state analysis of the triple tandem queue

Results are presented in Fig. 7. This experiment was also run three times; the values in the plots show the average of the convergence times measured. Case studies I–VI span along the x-axis of each plot.

Oil Pipeline. Consider a consecutive-k-out-of-n: F system ($\mathrm{C}(k, n : F)$). This consists of a sequence of n components *ordered sequentially*, so that the whole system fails if k or more *consecutive* components are in a failed state. For a more down-to-earth mental picture consider an oil pipeline where there are n equally spaced pressure pumps. Each pump can transport oil as far as the distance of k pumps and no further. Thus if $k > 1$, the system has certain resilience to failure and remains operational as long as no k consecutive pumps have failed.

Several generalizations exist to the original setting; we are interested in the non-Markovian and repairable systems analyzed in e.g. [27,34]. Those works assume the existence of a repairman which can take one failed component at a time and leave it "as good as new", after a log-normal distributed repair time has elapsed [34]. In particular [27] consider also the existence of non-Markovian failure times (namely, sampled from the Rayleigh—or Weibull—distribution) and measure the steady-state unavailability of the system.

We will limit here to the oil pipeline of the type $\mathrm{C}(5, 20 : F)$, i.e. where there is a total of $n = 20$ pressure pumps, and $k = 5$ consecutive failed pumps cause a general system failure. This was the most difficult case we run, where the estimated probability are in the order of 2.62e−9 and 7.49e−9 for the exponential and Rayleigh, case respectively. Other parameters are studied in [3]. In this

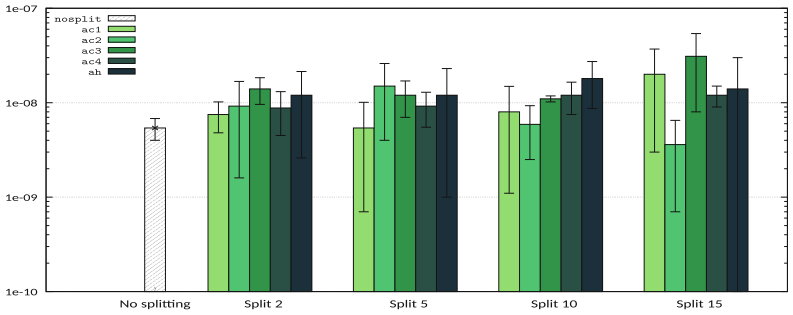

Fig. 8. Exponential-failures oil pipeline; intervals precision for 3 h timeout

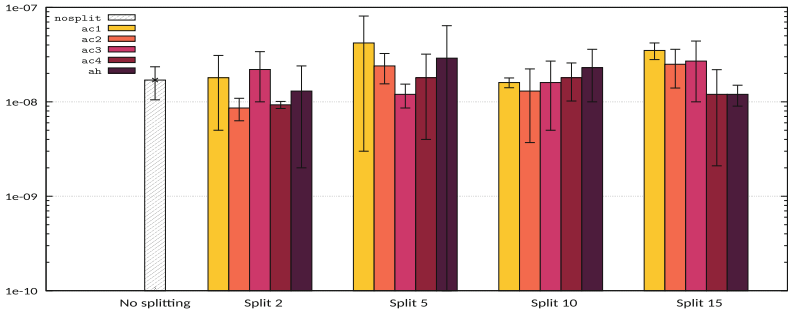

Fig. 9. Rayleigh-failures oil pipeline; intervals precision for 3 h timeout

setting, the steady-state system unavailability is given by the property query: $\mathsf{S}(\bigvee_{i=1}^{15}(b_i \wedge b_{i+1} \wedge b_{i+2} \wedge b_{i+3} \wedge b_{i+4}))$, where b_i indicates that component i is broken.

Also, we present a variation of the original model by changing the policy of repair, since the policy used in [27] is quite singular and cannot be modeled with FIG input language. Instead, we chose a priority policy where lower numbered components have more priority than higher numbered components.

The large number of components of this model prevents us to use the monolithic approach to derive the important function. Therefore the automatic importance functions tested are only compositional. The naïve strategy of composing the local functions with summation as composition operand is denoted `ac1`. Similarly, `ac2` uses product as composition operand and an exponentiation post-processing. Taking advantage that the propositional formula is in DNF, we use the $(\max, +)$ and $(+, *)$ semirings composition strategies and we denote them by `ac3` and `ac4` respectively. Last, `ah` implements an *ad-hoc* function with the $(\max, +)$ semiring, using the variables of the modules rather than the local importance functions which the tool could compute if requested. This is the approach followed in [27] and denoted $\Phi(t) \doteq cl - oc(t)$ in that work.

Due to the fact that this model takes too long to simulate, we decided to run it for 3 h and compare the resulting precision of the intervals for a confidence of 90%. We run three independent experiments. The results are presented in Figs. 8

and 9. These values are the average of the precision of the intervals obtained from the three experiments run; the deviation is shown as whiskers on top of the bars. We observe that, in this case study, the normal Monte Carlo was still competitive and postpone further discussions for the next section.

7 Concluding Remarks

In this paper we have reported on the continuation of the work on stochastic automata and its analysis that took place under Ed Brinksma's supervision during the late 90s [8,10–12]. We presented here a new variant of stochastic automata, named IOSA, amenable for simulation, and moreover, we reported on our efforts on obtaining a fully automatic implementation of the importance splitting technique for rare event simulation.

Our technique on automatically deriving importance function has proven highly competitive when compared with known good *ad-hoc* importance functions. This is evident in all experimental results reported in the previous section as well as in [3–5]. Yet, we know that we need to improve the FIG tool. Particularly, we need a better automatic construction of the thresholds where splitting is produced. We are currently using known techniques [6,7] that are not always producing good results when combined with our method of deriving importance function and the RESTART method. This is evident in the oil pipeline case study. We are currently busy on a new technique for the automatic derivation of thresholds that we expect to report soon.

References

1. Alur, R., Dill, D.L.: A theory of timed automata. Theor. Comput. Sci. **126**(2), 183–235 (1994). https://doi.org/10.1016/0304-3975(94)90010-8
2. Bolognesi, T., Brinksma, E.: Introduction to the ISO specification language LOTOS. Comput. Netw. **14**, 25–59 (1987). https://doi.org/10.1016/0169-7552(87)90085-7
3. Budde, C.E.: Automation of importance splitting techniques for rare event simulation. Ph.D. thesis. Universidad Nacional de Córdoba, Argentina (2017)
4. Budde, C.E., D'Argenio, P.R., Hermanns, H.: Rare event simulation with fully automated importance splitting. In: Beltrán, M., Knottenbelt, W., Bradley, J. (eds.) EPEW 2015. LNCS, vol. 9272, pp. 275–290. Springer, Cham (2015). doi:10.1007/978-3-319-23267-6_18
5. Budde, C.E., D'Argenio, P.R., Monti, R.E.: Compositional construction of importance functions in fully automated importance splitting. In: Puliafito, A., Trivedi, K.S., Tuffin, B., Scarpa, M., Machida, F., Alonso, J. (eds.) Proceedigns of VALUETOOLS 2016. ACM (2017). https://dx.doi.org/10.4108/eai.25-10-2016.2266501
6. Cérou, F., Del Moral, P., Furon, T., Guyader, A.: Sequential Monte Carlo for rare event estimation. Stat. Comput. **22**(3), 795–808 (2012). https://dx.doi.org/10.1007/s11222-011-9231-6
7. Cérou, F., Guyader, A.: Adaptive multilevel splitting for rare event analysis. Stoch. Anal. Appl. **25**(2), 417–443 (2007)

8. D'Argenio, P.R.: Algebras and automata for timed and stochastic systems. Ph.D. thesis. University of Twente, Enschede (1999)
9. D'Argenio, P.R., Katoen, J.P.: A theory of stochastic systems part I: Stochastic automata. Inf. Comput. **203**(1), 1–38 (2005)
10. D'Argenio, P.R., Katoen, J., Brinksma, E.: An algebraic approach to the specification of stochastic systems. In: Gries, D., de Roever, W.P. (eds.) PROCOMET 1998. IFIP Conference Proceedings, vol. 125, pp. 126–147. Chapman & Hall (1998)
11. D'Argenio, P.R., Katoen, J.P., Brinksma, E.: A compositional approach to generalised semi-Markov processes. In: Proceedings of WODES 1998, pp. 391–387. IEE (1998)
12. D'Argenio, P.R., Katoen, J., Brinksma, E.: Specification and analysis of soft real-time systems: quantity and quality. In: Proceedings of 20th RTSS, pp. 104–114. IEEE Computer Society (1999). https://doi.org/10.1109/REAL.1999.818832
13. D'Argenio, P.R., Lee, M.D., Monti, R.E.: Input/output stochastic automata. In: Fränzle, M., Markey, N. (eds.) FORMATS 2016. LNCS, vol. 9884, pp. 53–68. Springer, Cham (2016). doi:10.1007/978-3-319-44878-7_4
14. D'Argenio, P.R., Sánchez Terraf, P., Wolovick, N.: Bisimulations for non-deterministic labelled Markov processes. Math. Struct. Comput. Sci. **22**(1), 43–68 (2012)
15. Desharnais, J., Edalat, A., Panangaden, P.: Bisimulation for labelled Markov processes. Inf. Comput. **179**(2), 163–193 (2002)
16. Garvels, M.J.J., Van Ommeren, J.K.C.W., Kroese, D.P.: On the importance function in splitting simulation. Eur. Trans. Telecommun. **13**(4), 363–371 (2002). https://dx.doi.org/10.1002/ett.4460130408
17. Giry, M.: A categorical approach to probability theory. In: Banaschewski, B. (ed.) Categorical Aspects of Topology and Analysis. LNM, vol. 915, pp. 68–85. Springer, Heidelberg (1982). doi:10.1007/BFb0092872
18. van Glabbeek, R.J., Smolka, S.A., Steffen, B.: Reactive, generative and stratified models of probabilistic processes. Inf. Comput. **121**(1), 59–80 (1995)
19. Hahn, E.M., Hartmanns, A., Hermanns, H.: Reachability and reward checking for stochastic timed automata. ECEASST, vol. 70 (2014). http://journal.ub.tu-berlin.de/eceasst/article/view/968
20. Hermanns, H. (ed.): Interactive Markov Chains. LNCS, vol. 2428. Springer, Heidelberg (2002). doi:10.1007/3-540-45804-2
21. Jegourel, C., Legay, A., Sedwards, S.: Importance splitting for statistical model checking rare properties. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 576–591. Springer, Heidelberg (2013). doi:10.1007/978-3-642-39799-8_38
22. Kroese, D.P., Nicola, V.F.: Efficient estimation of overflow probabilities in queues with breakdowns. Performance Eval. **36**, 471–484 (1999)
23. L'Ecuyer, P., Le Gland, F., Lezaud, P., Tuffin, B.: Splitting techniques. In: Rare Event Simulation using Monte Carlo Methods, pp. 39–61. Wiley (2009). http://dx.doi.org/10.1002/9780470745403.ch3
24. Milner, R.: Communication and Concurrency. Prentice-Hall Inc., Upper Saddle River (1989)
25. Reijsbergen, D., de Boer, P.-T., Scheinhardt, W., Haverkort, B.: Automated rare event simulation for stochastic petri nets. In: Joshi, K., Siegle, M., Stoelinga, M., D'Argenio, P.R. (eds.) QEST 2013. LNCS, vol. 8054, pp. 372–388. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40196-1_31
26. Villén-Altamirano, J.: RESTART simulation of networks of queues with Erlang service times. In: Winter Simulation Conference (2009), WSC 2009, pp. 1146–1154 (2009). http://dl.acm.org/citation.cfm?id=1995456.1995616

27. Villén-Altamirano, J.: RESTART simulation of non-Markov consecutive-k-out-of-n: F repairable systems. Rel. Eng. Sys. Safety **95**(3), 247–254 (2010). https://dx.doi.org/10.1016/j.ress.2009.10.005
28. Villén-Altamirano, M., Martínez-Marrón, A., Gamo, J., Fernández-Cuesta, F.: Enhancement of the accelerated simulation method restart by considering multiple thresholds. In: Proceedings of 14th International Teletraffic Congress, pp. 797–810 (1994)
29. Villén-Altamirano, M., Villén-Altamirano, J.: RESTART: a method for accelerating rare event simulations. In: Queueing, Performance and Control in ATM (ITC-13), pp. 71–76. Elsevier (1991)
30. Villén-Altamirano, M., Villén-Altamirano, J.: The rare event simulation method RESTART: efficiency analysis and guidelines for its application. In: Kouvatsos, D.D. (ed.) Network Performance Engineering. LNCS, vol. 5233, pp. 509–547. Springer, Heidelberg (2011). doi:10.1007/978-3-642-02742-0_22
31. Villén-Altamirano, J.: Asymptotic optimality of RESTART estimators in highly dependable systems. Reliab. Eng. Syst. Saf. **130**, 115–124 (2014). www.sciencedirect.com/science/article/pii/S0951832014001227
32. Wolovick, N.: Continuous probability and nondeterminism in labeled transition systems. Ph.D. thesis. Universidad Nacional de Córdoba, Argentina (2012)
33. Wu, S., Smolka, S.A., Stark, E.W.: Composition and behaviors of probabilistic I/O automata. Theor. Comput. Sci. **176**(1–2), 1–38 (1997)
34. Xiao, G., Li, Z., Li, T.: Dependability estimation for non-Markov consecutive-k-out-of-n: F repairable systems by fast simulation. Reliab. Eng. Syst. Saf. **92**(3), 293–299 (2007). https://dx.doi.org/10.1016/j.ress.2006.04.004
35. Zimmermann, A., Maciel, P.: Importance function derivation for RESTART simulations of Petri nets. In: RESIM, pp. 8–15 (2012)