Rare Event Simulation of Stochastic Activity Networks Using Partition of the Region Technique

Amir Jalaly Bidgoly[*] Mohammad Abdollahi Azgomi[†]

Abstract

Modelling and evaluation is an important step of the development of complex systems. In most cases, discrete-event simulation is the only technique that can be used for this purpose. In highly dependable systems, simulation of rare events, corresponding to the failures of system components, will be quite time consuming. A number of techniques are proposed to speed-up simulation of rare events. This paper presents a novel approach for fast simulation of rare events modelled as *stochastic activity networks* (SANs). SANs are a stochastic extension of Petri nets. The solution is based on *partition of the region* (POR) technique and *stratified sampling* method. We have evaluated the proposed method using several examples. Results show that the simulation time is considerably decreased, comparing to the naïve simulation (up to 100 times) and to the *imporance sampling* technique. Moreover, the relative error of the simulation results is declined considerably. The proposed method is not dedicated to SANs and can be used for rare event simulation of the other extensions of Petri nets.

1 Introduction

Modelling and evaluation of real-world systems, needs analysing a large and complex model. Analytic techniques can be used to solve a wide range of such models, which are based on state space methods. State space generation is not possible for most models due to the state space explosion problem. In such cases, discrete-event simulation is the only possible technique.

In highly dependable systems that contain rare events, the cost of simulation will increase considerably. The main problem in simulation of highly dependable systems is the small probabilities associated to some important events. These small probabilities make the simulation to be run too long, because of the small times spent in these important rare events.

Rare event simulation is a key tool in some several areas such as reliability evaluation, telecommunications networks, switching systems and similar areas [3, 15, 16]. One famous approach for solving this problem is a technique called *importance sampling* (IS) [4, 12, 16] that is a Monte Carlo simulation variance reduction technique. In typical rare event setting, Monte Carlo method is not viable unless an *acceleration technique* is used to help rare events to occur more frequently. Another

[*] Performance and Dependability Engineering Laboratory, Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran, amir.jalaly@gmail.com

[†] Performance and Dependability Engineering Laboratory, Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran, azgomi@iust.ac.ir

A. Argent-Katwala, N.J. Dingle and U. Harder (Eds.): UKPEW 2008, Imperial College London, DTR08-9
pp. 107–122, http://ukpew.org/

acceleration frequency used for this purpose is *splitting* [11, 18, 37]. IS increases the probability of rare events by changing the probability laws to help the simulation to run faster. Also, it forces the simulation model to focus on rare events. Then, it multiplies the estimator in *likelihood ratio* to correct the result and get an unbiased estimator. The likelihood ratio is roughly the ratio of the original measure and the new measure associated with the generated path. In this way, the cost of evaluation will decrease and will become more acceptable. The main problem in general IS is to find a good probability law. This problem is defined in [15, 31, 4] as a good governor for the model. Unfortunately, a large part of rare-event simulation is focused on *static importance sampling* techniques. This means that a fixed change of measure is used throughout the simulation. While some literature is focused on to the *adaptive importance sampling* technique that changes the measure based on sample simulation path [2].

IS provides several methods that are different in efficiency and are suitable for various problems. Lewis and Bohm has tested IS on Markovian unreliability models and developed *failure biasing* and *forced transition* [17]. Later, Goyal et al. extended the above method in SAVE language [13]. SAVE is basically a generalised machine repairman model. Today several modelling tools provide IS [3031]. One of these tools is the *UltraSAN* [24, 25], which was used in 1990s for modeling and evaluation with *stochastic activity networks* (SANs) [20, 29]. SANs are a stochastic extension of Petri nets. These models have widely been used for performance and dependability evaluation in a wide range of systems.

L'Ecuyer and Tuffin has tried to improve IS by using *bounded relative error* (BRE) and *logarithmic efficiency* (LE) [18, 33, 34].

Splitting technique is also proposed as another approach for rare event simulation. *RESTART*‡ [36, 35, 37] is a techinque based on splitting. This method does not require changing the probability laws for acceleration, but an artificial drift toward the rare event is created by terminating with some probability

y trajectories that seems to go away from it and by splitting (cloning) those that are going in the right direction [19, 11, 9, 5]. The main idea in RESTART is repeation of important parts of the system (usually rare event) and getting a higher efficiency on these parts. This method is implemented in ASTRO [36]

This paper presents a novel idea for fast simulation of rare events in SAN models. The solution is based on *partition of the region* (POR) technique [27] as an extension of the *stratified sampling*. A variant of stratified sampling called *transition splitting* has been published in [10]. This technique is extremely efficient on models like an M/M/1 queuing model. The reason is that it uses all available exact knowledge and leaves very little to be simulated. It is very hard to find an appropriate transition splitting and to calculate the probability of each stratum.

Our new solution generally can be used for SAN models. The method is tested on four SAN models and the results and simulation time has been compared to the traditional discrete-event simulation. The results show that simulation is run in the shorter time length while the relative error of the results shows up to 100 times improvements.

The remainder of this paper is organized as follows: In section 2, some background theories are reviewed. In section 3, a new solution for simulation of SAN models

‡ *REpeative Simulation Trials After Reaching Thresholds*

based on POR technique is presented. In section 4, the results of four example models and analysis of the effect of the proposed method is presented. Finally, some concluding remarks are mentioned in section 5.

2 Background

In this section, we briefly review the background theories and techniques, which are used in the remainder of this paper.

2.1 Stratified Sampling

Let us consider the problem of estimating the below integral:

$$I = \int g(x)dx, \qquad x \in D \subset R^n \tag{1}$$

Let suppose that $g \in L^2(x)$ so $\int g(x)^2 dx$ exists and therefore that *I* exists. For stratified sampling must break the region *D* into *m* disjoint subregions D_i, *i* = 1, 2…, *m* that is:

$$D = \bigcup_{i=1}^{m} D_i, D_k \cap D_j = \varnothing, k \neq j \,.$$

Then let define:

$$I_i = \int_{D_i} g(x) f_X(x)dx, \tag{2}$$

I_i can be estimated separately.
The idea of this technology is similar to the idea of IS: simulation also take more observation in the parts of the region D that are more "important", but the effect of reducing the variance is achieved by concentrating more sample in more important subsets D_i, rather than by choosing the optimal probability density function (PDF).

2.2 Partition of the Region

This technique is similar to stratified sampling and may be able to present an extension for that [27]. In this technique we break the region D into two parts $D=D_1 \cup D_2$, representing the integral I defined in (1) as:

$$I = \int_D g(x)dx = \int_{D_1} g(x)dx + \int_{D_2} g(x)dx. \tag{3}$$

Let us assume that the integral:

$$I_1 = \int_{D_1} g(x)dx \tag{4}$$

can be calculated analytically. Let us define a truncated PDF as:

$$h(x)\begin{cases} \dfrac{f_X(x)}{1-P}, & if\ x \in D_2 \\ 0, & otherwise \end{cases} \tag{5}$$

where:

$$P = \int_{D_1} f_X(x)dx \tag{6}$$

By applying above formulas *h(x)* became an acceptable PDF. Formula (3) can be written as:

$$
\begin{aligned}
I &= I_1 + \int_{D_2} g(x)dx \\
&= I_1 + \int_{D_2} \frac{g(x)}{h(x)} h(x)dx \\
&= I_1 + E\left[\frac{g(X)}{h(X)}\right] \\
&= I_1 + (1-P)E\left[\frac{g(X)}{f_X(X)}\right]
\end{aligned} \tag{7}
$$

An unbiased estimated of I is:

$$
Y = I_1 + (1-P)\frac{g(X)}{f_X(X)} \tag{8}
$$

And the integral I can be estimated by:

$$
\theta = I_1 + (1-P)\frac{1}{N}\sum_{i=1}^{N}\frac{g(X_i)}{f_X(X_i)} \tag{9}
$$

This estimator can be used for simulation need some accelerator such as rare event simulation. The variance decrease depended on how break the region.

In section 3, we will try to introduce a method that breaks simulation area in a SAN model in such a way that all unimportant and important events divide into two separate parts. *Important events* are those events that take a role in computation of rare events. Similarly, *unimportant events* are those events that have not any role in computation of rare events.

2.3 Stochastic Activity Networks

Stochastic activity networks (SANs) [20, 29] are a general and stochastic extension of the Petri nets. SANs are powerful and flexible models for concurrent and distributed systems. These models are supported by several powerful modelling tools, such as *UltraSAN* or Möbius [8].

Elements of SANs are *places, gates* and *activities* as in Petri nets [20]. Places are same as Petri nets. Gates are used to connect places and activities and have two types: *Input* and *output*. Input gates connect one or more places to a single activity and have a *predicate* and a *function*. Output gates connect an activity to one or more places. These gates have only a function. The other element of SAN is activity. There are two types of activities: *timed* and *instantaneous*. Timed activities against instantaneous activities have a delay to *complete*. Delay time represented by a distribution function called *activity time distribution. F*(., μ; *a*) denotes the distribution function for activity *a* in marking μ. Activities also have *case*s in their outputs to show uncertainly about action taken at completing. *C*(., μ; *a*) used for showing *case distribution function* of activity *a* in marking μ.

An activity called *enabled* when all gates and places connected directly to activity hold *true*. The predicate of gates must return true and places must has at least one token. When an activity is enabled, it waits for completing. Time for delay taken from time activity distribution called *activity time.* After completion, first the input functions executed and then the output function. An enabled activity does not require completion. During activity time, SAN can move to a marking that activity be no more enabled and *aborted.*

3 Fast Simulation of SANs Using POR Technique

Consider a SAN model with possible path set Ω and possible events set *Π*. Our interest is in estimating a probability *P(ε)* of a rare event $\varepsilon \in \Pi$. Let *I(ε)* be an *indicator function*, *ε*, which is defined as follows:

$$\varepsilon = \begin{cases} 1 & \text{if the events belong to } \varepsilon \text{ are simulated in the model} \\ 0 & \text{otherwise} \end{cases}$$

In practice, a *reward variable* can be defined for the indicator function. Let γ denote the probability *P(ε)*. This may be estimated by Monte Carlo i.e. generating *n* independent sample of *I* ($I_1(\varepsilon), I_2(\varepsilon), \ldots, I_n(\varepsilon)$) and taking the average $\frac{1}{n}\sum_{i=1}^{n} I_i(\varepsilon)$ as estimator of *P(ε)* called γ_n. In general Monte Carlo method, when $P(\varepsilon) \to 0$ for reaching $\gamma_n \to \gamma$ almost sure as $n \to \infty$. This cause to relative error (or relative variance) be constant or at least be bounded [15, 18].

First try to solve this problem need to decrease variance (related error or related variance). For getting best result variance must be zero [16]. This needs *I(ε)* be equal to one in every simulation run. However, this is not possible to be implemented. In practice, this guides us to run the model in a way that the probability of executing rare events increases. One approach is using POR such that only the important part of model is simulated, which means rare events in a rare event simulation.

In this case first must find a way that break the path set (here a SAN model). Partitioning must keep all rare events in one part that is the goal of simulation. If we can break model such that the second part has no rare event (*I(ε)* always be zero) so no need to simulate this part. This help run time and efficiency get dramatic results.

Let us redefine Ω as follows:

$$\Omega = \{\text{all path in model} \mid \text{start state of path} = \text{final state of path} = x_0\} \tag{10}$$

where, x_0 is the initial state of the model.

In the next step, we need to break Ω set into two separate sets, as below:

$$\begin{aligned} \Omega_1 &= \{\chi \in \Omega \mid I(\varepsilon) = constant \text{ in } \chi\} \\ \Omega_2 &= \{\chi \in \Omega \mid \chi \notin \Omega_1\} \end{aligned} \tag{11}$$

which are our new state space sets.

Note that only the second set in important for us, which contains rare events. In practice, the place that is the beginning of Ω_1 and Ω_2 paths is chosen as the start state, i.e. the last common place that belongs to the intersection of the two sets.

In the next step, the SAN model will be partitioned into two separate parts. For this purpose, the first part must be divided into events set *Π*. Let us define two new sets as follows:

$$\begin{aligned} \Pi_1 &= \{e \in \Pi \mid \nexists \chi \in \Omega_2 \mid e \text{ is in } \chi\} \\ \Pi_2 &= \{e \in \Pi \mid \exists \chi \in \Omega_2 \mid e \text{ is in } \chi\} = \\ &\Pi - \Pi_1 \end{aligned} \tag{12}$$

In simple words if event *e* is being ran in simulation of a path *x*, *e* is a member of Π_2 if and only if there is some *x* that belong to Ω_2. Otherwise *e* is a member of Π_1. If find some $x_1 \in \Omega_1$ and $x_2 \in \Omega_2$ *e* will be in Π_2. So, common events between two path sets will be in Π_2 respectively.

Final step is computing *P* as defined in (6). Let us define a new indicator function:

$$I(\varepsilon') = \begin{cases} 1 & \varepsilon' \in \Pi_1 \\ 0 & otherwise \end{cases} \tag{13}$$

A simple simulation can evaluate $P(\varepsilon')$. ε' has no rare event and a general Monte Carlo method take result with good confidence level.

$$\hat{\gamma}_n(P') = \frac{1}{n}\sum I(\varepsilon'). \tag{14}$$

In practice, $I(\varepsilon')$ implement via a reward variable and it is large enough along simulation to compute in normal executing.

Then it is easy to break SAN model. Just remove all events belong to Π_1. All remained events are rare or need to run rare events. Definitely it is reduce variance to dramatic small value. In continue just use the below estimator:

$$\hat{\gamma}_n(P) = \frac{1}{n}\sum I(\varepsilon).(1 - P') \tag{15}$$

that take random variable only from important space of model Ω_2. It is clear that

$$\int_{\Omega_1} I(\varepsilon) d\varepsilon = 0 \tag{16}$$

This help remaining part simulate correctly respect to rare event evaluation.

Example. A SAN model is shown in Figure 1, which models a rare event with a low distribution of 0.0001. In this model Ω simply can be defined as

$$D = \{(rare-event, act1, act2), (act3, act4, act5)\}; \tag{17}$$

place1 has one token in the initial state and all other places are empty. So, initial state is [1, 0, 0, 0, 0]. Now let us consider indicator function point to token in *place3* that is a result of completing rare event activity. Ω_1 and Ω_2 respectively define as

$$\Omega_1 = \{(act3, act4, act5)\}$$
$$\Omega_2 = \{(rare-event, act1, act2)\} \tag{18}$$

So, Π_1 and Π_2 will be

$$\Pi_1 = \{act3, act4, act5\}$$
$$\Pi_2 = \{rare-event, act1, act2\} \tag{19}$$

It is clear that $I(\varepsilon')$ points to completing of *act3, act4, act5*. Also clear that P' simply can be compute.

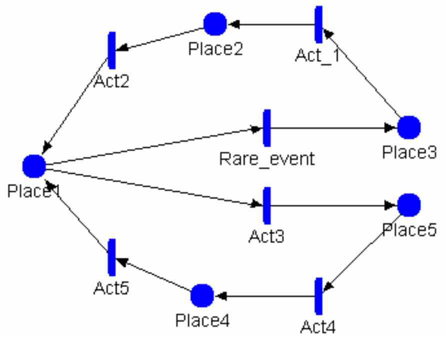

Figure 1. A SAN model with a rare event

Table 1. Activities of SAN model in Figure 1

Timed Activity	Distribution Parameters
Act3	Exponential 100
Act4	Exponential 10
Act5	Exponential 10
Act1	Exponential 10
Rare_Event	Exponential 0.0001

Just as last step for using POR we define probability P' as required in (15).
Let us define a new reward variable R' as

$$R = \begin{cases} \sum_{p \in N} mark(p) & reward - function \\ 0 & impluse - function \end{cases} \quad (20)$$

That p is a place in set {*place4, place5*} (an unimportant part) and *mark (p)* return number of token in p. For computing R' SAN model must simulate in normal mode. In this step fast simulation does not require, because R' does not contain any rare event. Result of R' is the value of probability P' defined in (15). An Impulse reward can be used for this purpose respectively. However we don't define that for simpler implementation.

As POR shows model breaks into two parts Ω_1 and Ω_2. Ω_2 is the important part and must observe more than Ω_1. Ω_1 can simulate normally But about Ω_2 that contain goal of simulation probability P' help us to estimate result in correct mode therefore we have a new SAN model that only contain set Ω_2 and Π_2. New SAN model of Figure 1 is shown in Figure 2.

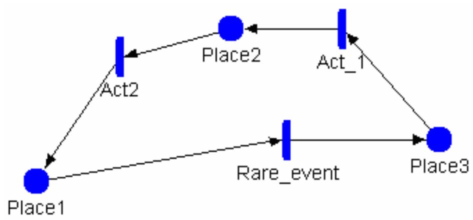

Figure 2. New SAN model of Figure 1 for using POR

4 Applications and Results

In this section we present the results of using our proposed method on four sample SAN models. To evaluate the proposed method, we have used the Möbius modelling tool. First, a simple SAN model with only one rare event is simulated. Then, another model with some other normal events is tested. Finally, a third model with two rare events and some interesting properties is tested. And finally, we have chosen a sample to compare the proposed method with the IS technique.

4.1 Example 1: A Simple Model

As the first example a simple SAN model is tested. This model contains only a rare event with a normal event that will race together. The SAN Model is shown in Figure 3 and its properties in Table 2 and Table 3.

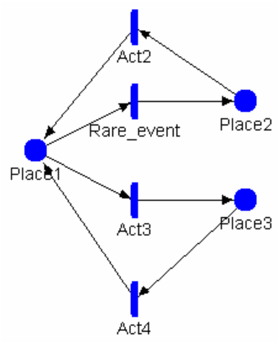

Figure 3. A simple SAN model with one rare event

Table 2. Intial markings of SAN model of Figure 3

Place Names	Initial Markings
Place1	1
Place2	0
Place3	0

In first step Ω_1 and Ω_2 must define:

$$\Omega_1 = \{(act3, act4)\}$$
$$\Omega_2 = \{(rare-event, act2)\} \quad (21)$$

And

$$\Pi_1 = \{act3, act4\}$$
$$\Pi_2 = \{rare-event, act2\} \quad (22)$$

Table 3. Activities of SAN model of Figure 3

Timed Activity	Distribution Parameters
Act2	Exponential 10
Act3	Exponential 100
Act4	Exponential 10
Rare_Event	Exponential 0.00001

In next step we define a reward variable as (23) for computing probability P'. This reward variable is simply defined on Ω_1 set.

$$R = \begin{cases} mark(place3) & reward-function \\ 0 & impluse-function \end{cases} \quad (23)$$

We evaluate this reward variable by Möbius modelling tool and get the result as 9.090908e-001. Now we test the original model and our new model, which contains only *Place1* and *Place2* for the result of the reward variable defined in (24) (i.e. removing *Act3* activity).

$$R = \begin{cases} mark(place2) & reward-function \\ 0 & impluse-function \end{cases} \quad (24)$$

Simply this new model is contains only Π_2 set Simulation results of this model for evaluation of rare event are shown in table 5. Simulator uses (1-P') to biased the estimator. Remember that goal is rare event or in the other word token in *place2*.

Simulator run both traditional and POR methods. The time is in terms of seconds. The results represent huge improvements in simulation time and precision. The results are compared to the outputs of the *steady state solver* of Möbius modelling tool. This helps us to get a real relative error that shown in last column.

Table 4. Results of simulation of SAN model represnted of Figure 3

Method	1-*P*	Time(*s*)	Results	Replications	Confidence Interval	Error
Naïve Simulation	-	7267.404	9.6558236535E-08	240636000	1.0887829753E-08	6.21%
POR	0.09090809	186.843	9.6199450981E-08	69213000	9.6065787077E-09	5.81%

4.2 Example 2: A More Complex Example

In this example, the model has some other normal events running with rare event, so model is more than just two simple parts. In this model still there is no event commonly in Ω_1 and Ω_2. The model presented in Figure 4 obviously has two separate parts:

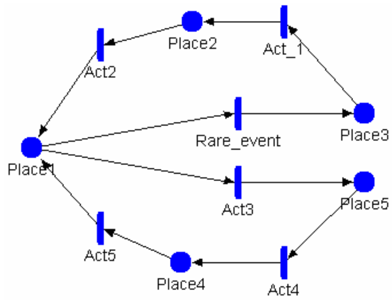

Figure 4. A SAN model with a rare event

Table 5. Results of simulation of SAN model represnted of Figure 4

Method	1-*P*	Time(*s*)	Results	Replications	Confidence Interval	Error
Naïve simulation	-	2012.609	5.0371920274E-07	67975000	5.0284734014E-08	5.78%
POR	0.04761813	42.922	4.6574525135E-07	12197000	4.6561591212E-08	2.19%

Table 6. Activities of SAN model of Figure 4

Timed Activity	Distribution Parameters
Act3	Exponential 100
Act4	Exponential 10
Act5	Exponential 10
Act1	Exponential 10
Rare_event	Exponential 0.0001

In the model presented in Figure 4, the initial markings of all places except *Place1* are zero. *Place1* has one token in its initial markings. The sets (Ω_1, Ω_2, Π_1, Π_2) of this model have shown in (18) and (19) so just see the result of simulation in table 6. Results in this test are also compared with outputs of steady state solver of Möbius software.

4.3 Example 3: Two Rare Events

As third example, we choose a model with two rare events. This model has a special property that make it distinguished from previous samples. In the presented model, Ω_1 and Ω_2 sets have some common events in their members. We interested in this model from this view that only just one activity is removed for fast simulation by our method however all places still cooperate in simulation. Notice that in previous sample some places goes away from simulation. The model is shown in Figure 5. The indicator function points to fail events so; the goal is computing the following reward variable:

$$R = \begin{cases} mark(fail1) + \\ mark(fail2) & reward-function \\ 0 & impluse-function \end{cases} \tag{25}$$

Sets Ω_1, Ω_2 are as below

$$\begin{aligned} \Omega_1 &= \{(job\_request, job\_done)\} \\ \Omega_2 &= \{(job\_request, fail1, repair1), \\ &\quad (job\_request, fail2, repair2)\} \end{aligned} \tag{26}$$

In the previous examples, the start state of paths is same as the start state defined for model, but in this model to simplify the paths we have changed the sets as bellow:

$$\begin{aligned} \Omega_1 &= \{(job\_done, job\_request)\} \\ \Omega_2 &= \{(fail1, repair1, job\_request), \\ &\quad (fail2, repair2, job\_request)\} \end{aligned} \tag{27}$$

It means that the start state is (*job_doing,* 1) marking (the last common state between two sets). In continue we define

$$\begin{aligned} \Pi_1 &= \{job\_done\} \\ \Pi_2 &= \{job\_request, fail1, fail2, repair1, repair2\} \end{aligned} \tag{28}$$

As seen in (28) *job_request* is common between two path sets however at last it moved to Π_2 set. Indicator of ε' points to completing of activity *job_done* and *job_request* in sequence. Simply because completing sequence *fail, repair, job_*request is a rare event, P can computing by supposing the all tokens in *idle* place is moved in by completing *job_done* activity. A reward variable can be defined for this purpose. Definition of this variable is as follows:

$$R = \begin{cases} mark(idle) & reward-function \\ 0 & impluse-function \end{cases} \tag{29}$$

Probability of P' is the value of evaluating of the variable in (29). However this reward variable is not actually as same as analytic definition of indicator of ε' but in practice they are such close that can be assume one. Now model can be simulated easily by removing *job_done* activity and using 1-P' probability for biasing the estimator. Result of this simulation is shown in table (7). This model is compared with results of steady-state solver of Möbius modelling tool.

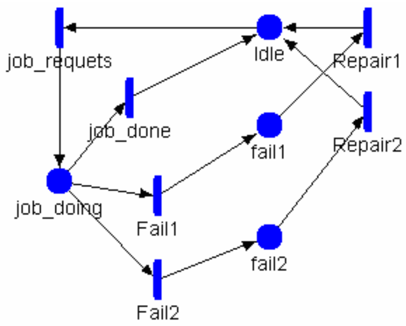

Figure 5. A SAN model with two rare events

Table 7. Intial markings of SAN model of Figure 5

Place Names	Initial Markings
idle	1
Fail1	0
Fail2	0
Job_doing	0

Table 8. Activity of SAN model of Figure 5

Timed Activity	Distribution Parameters
Job_request	Exponential 100
Job_done	Exponential 1000
Fail1	Exponential 0.0001
Fail2	Exponential 0.00001
Repair1	Exponential 1
Repair2	Exponential 10

Table 9. Results of simulation of SAN model represnted of Figure 5

Method	1-*P*	Time(*s*)	Results	Replications	Confidence Interval	Error
Naïve simulation	-	2718.750	8.7378789281E-06	13210000	8.7044577598E-07	4.83%
POR	0.09092855	788.547	9.1541250070E-06	100000000	2.0061251233E-07	0.29%

4.4 Example 4: POR vs. IS

For last example of presented method, we test that on a model studied by Obal II and Sanders [26] for IS technique presenting in *UltraSAN*. This model can be seen in Figure 6. Obal II study the unreliability of this model over an interval of time. This can be computed through an *instant of time reward variable* when the system failure marking is an absorbing marking by examining the instantaneous rate reward at the end of interval [26]. Model is a *machine-repairman system* that uses a delayed group repair policy. There are two types of components in the system, with different failure rates. The places labeled type 1 and type 2 model the two types of components. The marking of each place represents the number of working components of that type. In this case, there are two *type-one* components, and four *type-two* components. Timed activities *fail_1* and *fail_2* model the time between failures for each component. As shown in Table 11, the failure times are exponentially distributed with marking dependent rate parameters. A component of type-one fails with rate 0.005, while components of type-two fail at twice that rate. The use of marking dependent rate parameters allows us to avoid including an activity for each component's failure time distribution, resulting in a more compact representation.

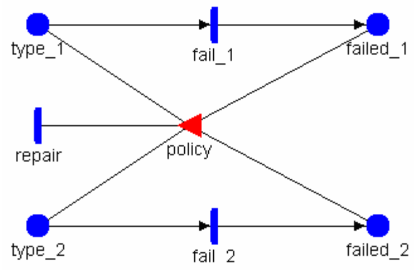

Figure 6. The *machine-repairman* model presented by Obal II

Table 10. Intial markings of SAN model of Figure 6

Place Names	Initial Markings
type_1	2
type_2	4
failed_1	0
failed_2	0

Table 11. Activity of SAN model of Figure 6

Timed Activity	Distribution Parameters
fail_1	Exponential 0.005*Mark(*type_1*)
fail_2	Exponential 0.01*Mark(*type_2*)
repair	Exponential 1

Table 12. Activity of SAN model of Figure 6

Gate	Enable Predicate	Function
policy	(type_1->Mark() > 0 || type_2->Mark() > 0) && (failed_1->Mark() == 2 || failed_2->Mark() > 1)	if (failed_1->Mark() == 2) { failed_1->Mark()=0; type_1->Mark() = 2; } else { failed_2->Mark() = 0; type_2->Mark() = 4; }

The markings of places *failed_1* and *failed_2* represent the number of components of each type that have failed. There is one repairman in the system. The repair policy is to wait until at least two components of the same type have failed, and then begin to repair the whole group. Type-one component repair is given preemptive priority over repair of type-two components. When the repair is completed, all components of that type are as good as new. This repair policy is implemented in the input gate policy. The properties of *policy* are shown in table 12. *type_1*->Mark() points to marking of *type*_1 and so on. If all components of both types fail, the system fails, and all repair activity halts; the failed state is an absorbing state.

For evaluation of unreliability of model a reward variable is used. A reward structure that identify the failed state is

$$R = \begin{cases} \begin{cases} 1 & if\ (type\_1,0) and (type\_2,0) \\ 0 & otherwise \end{cases} & reward-function \\ \quad 0 & impluse-function \end{cases} \tag{30}$$

Goal is computing the unreliability in interval [0,100]. This can be done by using reward variable as instant of time reward variable.

For using this model in POR method, we must break it into two parts. For this purpose we define two type of repair: *repair_type_1* and *repair_type_2*. Each repair type note to repairing of one type e.g. *repair_type_1* is repair activity on type one components. Also as described before the start state for defining cycle is the last marking common between two parts. This state simply is {(*failed_1*, 2), (*failed_2,* 4)} when system fails and {(*failed_1*, 2)} or {(*failed_2,* 2)} when *repair* enabled. So sets Ω_1, Ω_2 are as below

$$\begin{aligned} \Omega_1 &= \{(repair-type\_1, fail1, fail1), (repair-type\_2, fail2, fail2), ...\} \\ \Omega_2 &= \{(systemfail)\} \end{aligned} \tag{31}$$

Ω_1 members are those who run *repair* activity however Ω_2 has only one member that is fail state. If we continue this partitioning, for computing *P* we must evaluate following probability when a component fail.

$$1-P = \frac{P(\Omega_2)}{P(\Omega_1)+P(\Omega_2)} = \frac{P(systemfail)}{P(systemfail)+P(repair)} \tag{32}$$

This probability is a rare event and is same as finding unreliability. So in this model we use some other Ω sets. These new sets however does not complete push rare events on one part and other events in other part, help simulation in a way to reduce time and variance. New sets defined as

$$\begin{aligned} \Omega_1 &= \{(repair-type\_1, ...)\} \\ \Omega_2 &= \{(systemfail), (repair-type\_2, ...)\} \end{aligned} \tag{33}$$

This partitioning breaks system into one part that contain repairing of type one components and another part has fail state and repairing of type two components. Now 1-*P* define as

$$1-P = \frac{P(\Omega_2)}{P(\Omega_1)+P(\Omega_2)} = \frac{P(systemfail)+P(repair-type\_2)}{P(systemfail)+P(repair-type\_1)+P(repair-type\_2)} \tag{34}$$

Probability (32) can evaluate simply. Partitioned model has no repair activity for type one components. This increase the frequently of system fail while (1-*P*) probability help for biasing the estimator. The result of this simulation is shown in table 14. Results of Obal II simulation also present in table 13.

Table 13. Results of simulation of SAN model represnted of Figure 6

Method	Time(*s*)	Results	Replications	Error
Naïve simulation	67230	1.52E-06	28375000	20.8%
IS	1385	1.90E-06	338497	1.02%

Table 14. Results of simulation of SAN model represnted of Figure 6

Method	1-*P*	Time(*s*)	Results	Replications	Error
Naïve Simulation	-	1290.203	1.7176054559E-06	79180000	10.5%
POR	0.1147283	73.203	1.9193768592E-06	8189000	0.01%

Both simulation (naïve and partitioned) run with 98% confidence level. Result in our simulation show about 17 times improvement in simulation time and about 1050 times in relative error while Obal II's results shows 48 in former and 20 times in later, respectively. Note that in this model, POR is not used completely because of reaching another rare event when computing the probability *P*. The problem of using the presented method for this model is that common part between Ω_1, Ω_2 is not so much so computing *P* is need computing original rare event. However, Simulation results' show good improvements.

5 Conclusions

Partition of the region by breaking the region of the simulation into two parts helps simulator to spend more time on important parts of the model. In this paper, using this technique a new approach for fast simulation of SAN models is introduced. For this purpose, we have defined sets Ω_1 and Ω_2 than breaking a model using Π_1 and Π_2 sets. Then, the probability of being in the important part that is defined by a new indicator function is computed. And finally, the model is simulated without events in set Π_2.
Since partition of the region uses simulation in a way that only rare events are observed, it improves efficiency of simulation. Also this technique can simply define dynamically so one can enjoy this method automatically on every SAN model.
We have evaluated the proposed method using four examples of SANs. Results show that simulation time is decreased even up to 100 times, while the results of the simulation show high improvements regarding the relative errors.
The method presented in this paper is not dedicated to SANs and can also be used with other stochastic extensions of Petri nets, such as SPNs, GSPNs, etc. It is also possible to use this method with Markov chains. We are currently working to use the proposed method for rare event simulation of SPNs and Markov chains.

References

1 Blum, A.M. Heidelberger, P. Lavenberg, S.S. Nakayama, M. and Shahabuddin, P., "System availability estimator (SAVE) language reference and user manual version 4.0," Technical Report RA 219 S. IBM Thomas J. Watson Research Center, June 1993.

2 Borkar, V. S., S. Juneja, A. A. Kherani. "Performance analysis conditioned on rare event: An adaptive simulation scheme." Communication in Information, 3, 4, 259-278, 2004.

3 Bulhuis, P. G., Chandler, D., Dellago, C., and Geissler P. L., "Transition path Sampling: Throwing robe over rough mountain passes, in the dark," Annual Review on physical Chemistry 53, 2002, pp. 291-318.

4 Bucklew, J. A., *Introduction to Rare Event Simulation*, Springer-Verlag, New York, 2004.

5 Ce'rou, F., F. LeGland, P. Del Moral, and P. Lezaud, "Limit theorems for the multilevel splitting algorithm in the simulation of rare events", In Proc. of the 2005 Winter Simulation Conference, ed. F.B.A.M.E. Kuhl, N.M. Steiger and J.A. Joines, 2005, pp. 682–691.

6 Dellago, C. Bolhuis, P. G. Csajka, F. S. and Chandler, D., "Transition Path Sampling and the Calculation of Rate Constants," J. Chem. Phys. 108, 1998.

7 Dellago, C. Bolhuis, P. and Geissler. P. L., "Transition Path Sampling," Adv. Chem. Phys 123, 2002.

8 Deavours, D.D., et al., "The Möbius Framework and Its Implementation," IEEE Trans. on Soft. Eng. 28(10), IEEE CS Press, 2002, pp. 956-969

9 Garvels, M. J. J., *The splitting method in rare event simulation*, Ph. D. Thesis, Faculty of Mathematical Science, University of Twente, The Netherlands, 2000.

10 Gavioronski. A. A. "Transition splitting for estimation of rare events with application to high speed data networks," In Labatoulle and Roberts, 32, 1994, pp. 767–776.

11 Glasserman, P., P. Heidelberger, and P. Shahabuddin, "Asymptotically optimal importance sampling and stratification for pricing path dependent options," Mathematical Finance 9 (2), 1999, pp. 117–152.

12 Glynn, P. W. and Iglehart, D. L. 1989. "Importance sampling for stochastic simulations." Management Science 35, 1367–1392.

13 Goyal, A. Shahabuddin, P. Heidelberger, P. Nicola, V.F. and Glynn, P. W. "A Unified Framework for Simulating Markovian Models of Highly Dependable Systems," IEEE Transactions on Computers, vol. 41, no. 1, Jan. 1992, pp. 36-51.

14 Heegaard, P. E., "Speed-up techniques for simulation," Telektronikk, 1995.

15 Heidelberger, P., "Fast Simulation of Rare Events in Queuing and Reliability Models," ACM Transactions on Modelling and Computer Simulation, vol. 5(1), 1995, pp. 43-85.

16 Juneja, S. and Shahabuddin, P., "Rare event simulation techniques: An introduction and recent advances." In Simulation S. G. Henderson and B. L. Nelson, Eds. Handbooks in Operations Research and Management Science, Elsevier, Amsterdam, The Netherlands, 2006, pp. 291-350

17 Lewis, E. E. and Bohm, F., "Monte Carlo of Markov Unreliability Models," Nuclear Engineering and Design 77, pp. 49-62.

18 L'Ecuyer, P., Blanchet, J., Tuffin, B., and Glynn, P. W., "Asymptotic Robustness of Estimators in Rare event Simulation", ACM Transactions on Modelling and Computer Simulation, 2007.

19 L'Ecuyer, P., Demers, V., and Tuffin, B. "Splitting for Rare event Simulation," In Proc. of the 2006 Winter Simulation Conference, IEEE Press, 2006, 137-148.

20 Meyer, J.F. Movaghar, A. and Sanders, W.H., "Stochastic Activity Network: Structure, Behavior and Application," In Proc. of Int'l Workshop on Timed Petri Nets, Torino, Italy, July 1985, pp. 106-115.

21 Nicola, V.F. Nakayama, M. K. Heidelberger, P. and Goyal, A., "Fast Simulation of Dependability Models with General Failure, Repair and Maintenance Process," In Proc. of 20th Annual International Symposium on Fault-Tolerance Computing, Newcastle upon type, United Kingdom 1990, pp. 491-498.

22 Nicola, V.F. Shahabuddin, P. and Nakayama, M., "Techniques for Fast Simulation of Models of Highly Dependable Systems," In Proc. of the IEEE Transition on Reliability, Vol. 50, No 3, 2001.

23 Nicola, V.F. Shahabuddin, P. Heidelberger, P. and P.W. Glynn, "Fast simulation of Steady-State Availability in Non-Markovian Systems," 1993.

24 Obal II, W.D. and Sanders, W.H., "An Environment for Importance Sampling Based on Stochastic Activity Networks," In Proceedings of the 13th Symposium on Reliable Distributed Systems, Dana Point, CA, Oct. 1994, pp 64-73.

25 Obal II, W. D. and Sanders, W. H., "Importance Sampling Simulation in UltraSAN", From Simulation, Vol. 62, No. 2, Feb. 1994, pp. 98-111.

26 Obal II, W.D., "Importance Sampling Based of SAN-Based Reward Models" Master Thesis, The University of Arizona, July 1993.

27 Rubistein, R. Y., "Simulation and Monte Carlo Method," Wiley series in probability and mathematical statistics, John Wiley and sons, New York, United State of America, 1981, pp. 114-158.

28 Sanders, W. H., and Meyer, J. F., "A Unified Approach for Specifying Measures of Performance, Dependability, and Performability," in Dependable Computing for Critical Applications, Dependable Computing and Fault-Tolerance Systems, Vol. 4., A. Avizienis and J. C. Laprie, editors, Springer Verlag, Vienna, 1991, pp. 215-238.

29 Sanders, W. H., Meyer, J. F., "Stochastic Activity Networks: Formal Definitions and Concepts," Lectures on Formal Methods and Performance Analysis, 2001.

30 Shahabuddin, P., "Simulation and Analysis of Highly Dependable Systems," PhD Thesis, Stanford University, 1990.

31 Shahabuddin. P., "Importance Sampling for the Simulation of Highly Reliable Markovian Systems," management Science, vol. 41, no. 3, March 1994, pp. 333-352.

32 Shreider, Yu. A., "The Monte Carlo Method (the Method of Statistical Trials)," Pergamon, Elmsford, New York, 1966.

33 Tuffin, B., "Bounded normal approximation in simulations of highly reliable Markovian systems." Journal of Applied Probability 36, 4, 1999, pp. 974–986.

34 Tuffin, B., "On numerical problems in simulations of highly reliable Markovian systems," In Proc. of the 1st International Conference on Quantitative Evaluation of SysTems (QEST). IEEE CS Press, University of Twente, Enschede, The Netherlands, 2004, pp. 156–164.

35 Ville`n-Altamirano, M., and Ville`n-Altamirano, J., "RESTART: A Straightforward Method for Fast Simulation of Rare events," In Proceedings of the 1994 Winter Simulation Conference, 1994, pp. 282-289.

36 Ville`n-Altamirano, M., and Ville`n-Altamirano, J., "RESTART: A Method for Accelerating Rare Event Simulations," in J. W. Cohen, C.D. Pack, editors, Queuing, Performance and Control in ATM, 13th International Teletraffic Congress, Copenhagen, North-Holland,1991, pp. 71-76.

37 Ville`n-Altamirano M. and Ville`n-Altamirano, J. "On the efficiency of RESTART for multidimensional systems." ACM Transactions on Modeling and Computer Simulation 16, 2006, 3, 251–279.

38 Williamson, A., "Discrete Event Simulation in the Möbius Modelling Framework." Master's thesis, University of Illinois at Urbana-Champaign, 1998.