A general and efficient framework for improving Balanced Failure Biasing

Shijian Mao
Shanghai Key Laboratory of Trustworthy Computing
East China Normal University
Shanghai, China
shmaoshijian@gmail.com

Min Zhang*
Shanghai Key Laboratory of Trustworthy Computing
East China Normal University
Shanghai, China
mzhang@sei.ecnu.edu.cn

Jia Yan
Shanghai Key Laboratory of Trustworthy Computing
East China Normal University
Shanghai, China
51164500258@stu.ecnu.edu.cn

Yao Chen
Shanghai Key Laboratory of Trustworthy Computing
East China Normal University
Shanghai, China
51184501098@stu.ecnu.edu.cn

***Abstract*—Balanced Failure Biasing (BFB) is a way to simulate the probability of reaching a rare goal state in highly reliable Markovian systems (HRMSs). BFB gives the same probability to each ralely-arrived path of one state, therefore leading to large expenditures on paths with little influence on results. We propose a new framework using Stratified Sampling, which is a general and efficient framework for improving BFB. We introduce Stratified Sampling on BFB (SBFB), which divides the original state space into many subspaces, and rearranges the attention on each subspace. To make a further reduction on average path length, we introduce Stratified Sampling on Distance-based BFB (SBFB-D). According to experiments based on case of Workstation Cluster and case of Distributed Database System, SBFB has about 0.07% and 2.13% relative error on these two cases respectively, while SBFB-D has about 0.07% and 0.197%, comparing to standard BFB's 11.1% and 11.1%. Besides, SBFB spends about 12.30s and 28.65s on path simulation respectively, while SBFB-D spends about 13.10s and 17.40s, comparing to standard-BFB's 26.44s and 36.78s.**

***Index Terms*—Rare-event simulation , Stratified Sampling , Balanced Failure Biasing.**

I. Introduction

Increasing demand for system's reliability has created increased interest in fault-tolerant systems. However, realistic models of these fault-tolerant systems always have very large state spaces and ralely arrive at failure rates to the fault state , which leads to the demand of both time, space and accuracy. Numerical methods for evaluating these fault-tolerant systems such as those implemented in the model checking tool PRISM [1] proved to be computationally infeasible when facing with large state space.

In that case, Monte Carlo simulation on Markov chains would be a balance on time, space and accuracy, which sacrifices only a little accuracy and saves both time and space. It only requires the state transition conditions of the state space, and is therefore largely independent of its state space size. so we could deal with the highly reliable Markovian systems (HRMSs), which formalized the low probability failure paths as powers of some rarity parameter . With the HRMSs, we can estimate many fault-tolerant system problems, such as the mean time to failure, the unreliability, and some other demand that involve rare events.

The importance sampling technique [2] could be used to speed up standard simulation when it meets rare events for the reason of little probability of failure paths. The main idea behind importance sampling is to modify the sampling distributions so that those little probability failure paths could be sampled with higher probabilities, and thus gain a better accuracy with the same sampling numbers.

Failure biasing (FB) is an importance sampling scheme which was first proposed in [3], [4] for the simulation of the unreliability. This technique has been then adapted in [5], [6] for the simulation of steady-state unavailability, in [7] for mean time to failure, and in [8] for other dependability measures.

Balanced Failure biasing (BFB) was proposed in [9], which is a refined version of failure biasing. It meets the property of Bounded Relative Error (BRE), which means the relative error of an estimator remains bounded when the result need estimating goes to zero.

Distance Based BFB have been proposed in [10] called failure distance biasing, which uses the idea of single source shortest path like Dijkstra algorithm. Unbalanced systems has been extended in [11] by the same author, and were named failure transition distance biasing (FTDB) and balanced failure transition distance biasing (BFTDB).

High probability cycles (HPCs) would influence the per-

* The corresponding author.
Supported by the NSFC Project (No. 61672012).

978-1-7281-8915-4/20/$31.00 ©2020 IEEE
DOI 10.1109/QRS-C51114.2020.00081

formance of BFB by large amounts of loops in the cycles with high probability. So the implementable general biasing scheme (IGBS) was proposed in [12] to mitigate the effects of HPCs. Path-ZVA was proposed in [13], and they solved the HPCs problem on distance based BFB, which is a huge improvement.

In this paper, we focus on the Stratified Sampling framework [14], [15], which can be used to reduce variance, and we use it to improve many works on BFB. We divide the state space into many subspaces firstly, by stepping into the state space with k steps from initial states. With the k steps we can achieve many new states, and these new states can be seen as initial states of these subspaces. For these subspaces, approximate result can be simulated by a few samples, for instance, 5% of number of sampling you want to use in all. Then we can know the probability from initial state to subspaces' initial state to goal state, and then rearranging the number of sampling to each subspace according to its importance. According to our experiments, this framework can reduce the run time mainly because of its attention on important subspaces and shorter path length of subspaces with larger influence.

This Stratified Sampling framework is:

1) General, as it can be used on many different works what are based on BFB, and variance of result can be reduced.
2) Efficient, as it could pay more attention on more influential subspaces to reduce variance, and shorter average path length could save times of sampling. Further more, parallel computing is also supported.

The remainder of this paper is as follows. After a formal description of the Stratified Sampling framework in Section II, we describe how to nest it on BFB algorithms in Section III. Next, we would present an empirical evaluation of these techniques in Section IV, and make final conclusions in Section V.

II. Preliminaries

The model is given in a Markov chain with a state space χ, and we assume that the system starts at a unique initial state $s_0 \in \chi$ with a single goal state $g \in \chi$. The complete transition probability structure in the DTMC is given by p_{xz} of jumping from state x to state z, with $x, z \in \chi$. And the probability could be calculated through $p_{xz} = \epsilon^{r_{xz}}$, where ϵ is the rarity parameter like 0.01, and the product of two transitions could be calculated as

$$p_{xy} \cdot p_{yz} = \epsilon^{r_{xy}} \cdot \epsilon^{r_{yz}} = \epsilon^{r_{xy}+r_{yz}} \tag{1}$$

Let a path ω be a sequence $\omega(0), \omega(1), ..., \omega(n_\omega)$ of states in χ, with n_ω denoting the number of steps in the path. Let $\Omega(x)$ be the set of path with $\omega(0) = x$. For all $x \in \chi$ we define the probability that the rare event occurs, starting in x, as

$$\pi(x) \triangleq \sum_{\omega \in \Omega(x)} \mathbb{P}(w), \text{ where } \mathbb{P}(\omega) \triangleq \prod_{i=1}^{n_\omega} p_{\omega(i-1)\omega(i)} \tag{2}$$

The basic idea of evaluating π is through a point estimate: Draw $N \in \mathbb{N}$ sample pahs to obtain a sample set $\{\omega_1, ..., \omega_N\}$. For drawing a sample path, we start at s_0 and get successor states using $\mathbb{P}$ until we reach g. Let $1_\Omega(\omega)$ denote an indicator function which equals 1 if ω in Ω and 0 otherwise. This allows us to obtain an unbiased estimator of π, given by

$$\hat{\pi}_{\mathbb{P}} = \frac{1}{N} \sum_{k=1}^{N} 1_\Omega(\omega_k) \tag{3}$$

And approximate 95%-confidence interval for π can be obtained using the Central Limit Theorem.

Importance sampling is a technique which can ensure simulating using different transition probabilities q_{xz}. Let $\mathbb{Q}$ be the probability measure on paths defined analogously to $\mathbb{P}$ but for q_{xz}. We compensate for overestimation by weighting each outcome with the ratio of $\mathbb{P}$ and $\mathbb{Q}$. Every time a transition is sampled using the new probabilities, this weighting factor needs to be fixed. Our new estimator then becomes

$$\hat{\pi}_{\mathbb{Q}} = \frac{1}{N} \sum_{k=1}^{N} L_{\mathbb{Q}}(\omega_k) \cdot 1_\Omega(\omega_k), \text{ with } L_{\mathbb{Q}}(\omega) = \prod_{i=1}^{n_\omega} \frac{p_{\omega(i-1)\omega(i)}}{q_{\omega(i-1)\omega(i)}} \tag{4}$$

This estimator is unbiased for any new distribution by the Radon-Nikodym Theorem. In the following, we will write $\hat{\pi} = \hat{\pi}_{\mathbb{Q}}$ for brevity.

Thus, BFB uses the importance sampling, increases the probability of failure paths with symbol $\mathbb{P}^F$, and decrease the probability of these repair paths by $\mathbb{P}^R$ .Beside, $\mathbb{P}^C$ represents the probability of transitions having Self-circulating . Then, BFB increases the probability of failure paths as follows, where $I(p^F) = 1$ if $p^F > 0$ otherwise 0:

$$q_{xy} = \begin{cases} \theta \cdot \frac{I(p^F_{xy})}{\sum_{z:p^F_{xz}>0} I(p^F_{xz})} \\ p^C_{xy} \\ (1-\theta) \cdot \frac{p^F_{xy}}{\sum_{z:p^F_{xz}>0} p^F_{xz}} \end{cases} \text{, if } x \neq g \tag{5}$$

Besides, if using the Zero Variance Approximation (ZVA) approach as follows, the estimator would have zero variance when $v(z)$ could exactly equal to $\pi(z)$:

$$q_{xz} \triangleq \frac{p_{xz} v(z)}{\sum_{x' \in \chi} p_{xx'} v(x')} \text{ , where } v(z) \text{ is approximation for } \pi(z) \tag{6}$$

Here we could use Distance-based techniques to estimate the $v(z)$ with a lower variance, sacrificing the space complexity.

III. The Stratified Sampling Algorithm

In this section, we would describe two versions of the simulation mathod : Stratified BFB (SBFB) and Stratified distanced-based BFB(SBFB-D). In the following, we first give a formal description of these two methods, and then their implementations.

TABLE I
LIST OF SYMBOLS

χ	State space of the Markov chain
s_0,g	Initial state and goal state respectively
$\mathbb{P}$,p_{xy}	Original probability of the transition from state x to state y
$\mathbb{Q}$,q_{xy}	New probability of the transition from state x to state y
ϵ	The rarity parameter, which is used as ϵ-order of the transition
r_{xy}	ϵ-order of the transition from state x to state y,i.e.,$p_{xy} = \Theta(\epsilon^{r_{xy}})$
ω	A path, i.e., a sequence of states $\omega(0), \omega(1), ..., \omega(n_\omega)$
$\Omega(x)$	Set of all paths ω starting at $\omega(0) = x$
$\pi(x)$	$\sum_{\omega \in \Omega(x)} \mathbb{P}(\omega) = \mathbb{P}(\Omega(x))$
$\hat{\pi}_\mathbb{P}, \hat{\pi}_\mathbb{Q}$	Verification result from $\mathbb{P}$ and $\mathbb{Q}$ respectively
θ	Parameter of BFB to control balance between failure paths and the rest.
$v(x)$	Approximation of $\pi(x)$ by Distance-based algorithm
k	k steps forward from the initial state s_0
$\Pi(k)$	State set after k steps forward from state s_0
π_i	True result of the subspace i
$\hat{\pi}_i$	Verification result of the subspace i
$p(\pi_i)$	Probability to achieve subspace i from state s_0
$z_{1-\alpha}$	$(1-\alpha)$ quantity of the standard normal distribution
N	Sampling number which would be allocated to simulating the system
N_i	Sampling number which would be allocated to simulating the subspace i

A. Stratified Sampling on Standard BFB (SBFB)

Our method is to allocate a suitable number of sampling for each subspace, and a more important subspace needs more number of sampling. At the same time, lots of transition path could be saved. So the most important is the way to split.

Here we use the idea of breadth-first search: step from the initial state s_0, step forward k steps, and gain the achieved state set $\Pi(k)$, with the formal definition as follows, where g is the goal state,

$$\Pi(k) = \{\omega(k) : \omega \in \Omega(s_0), w(k) \neq g\} \tag{7}$$

And $\Pi(k)$ would be symbolized as K^- in the following paper, with $K^+ = \{x | x \notin K^-, \forall x \in \chi\}$, and $K^* = x | p_{xy} > 0, \forall x \in \omega(k), \exists y \in K^+$ is the boundary states between K^+ and K^-.

Besides, we define the simulation result of subspace i with $\hat{\pi}_i$, $i = 1, 2, ...m$, and m is the size of subspaces. While the Probability to each state in $\Pi(k)$ is obviously to calculate, which is formalized as $p(\pi_i)$. So the final result of simulation would be

$$\hat{\pi} = \sum_{i=1}^{m} p(\pi_i) \cdot \hat{\pi}_i \tag{8}$$

The $1-\alpha$ Confidence interval is as follows, where $z_{1-\alpha}$ is $1-\alpha$ quantity of the standard normal distribution, N_i is the number of sampling allocated to this subspace, and $\hat{\sigma}_i^2$ is variance of the i-th sample.

$$(\hat{\pi} - z_{(1-\alpha)/2} \sqrt{\sum_{i=1}^{m} \frac{p(\pi_i)^2 \cdot \hat{\sigma}_i^2}{N_i}}, \hat{\pi} + z_{(1-\alpha)/2} \sqrt{\sum_{i=1}^{m} \frac{p(\pi_i)^2 \cdot \hat{\sigma}_i^2}{N_i}}) \tag{9}$$

However, allocating the number of each subspace is a hard work. We here simulate each subspace by BFB with only a few samples, such as 5% of all, and allocate the number by the following approach [16], which can reach the smallest variance for $\hat{\pi}$:

$$N_i = N \cdot (\frac{p(\pi_i)\sigma_i}{\sum_{k=1}^{m} p(\pi_k)\sigma_k}) \tag{10}$$

$$Var(\hat{\pi}) = \frac{1}{N} (\sum_{i=1}^{m} p(\pi_i)\sigma_i)^2 \tag{11}$$

The algorithm for SBFB is described in Algorithm 1. k steps have been done in lines 2-9, for putting $p(\pi_i)$ into π_i's subspace s' where s' could be seen as key of subspaces. From lines 11-13 call BFB for the first time for each subspace for obtaining σ_i, and lines 14-17 use σ_i to allocate the number of sampling for BFB. Finally, line 18 merges all results from subspaces, and receives the great result.

This SBFB algorithm has almost the same time and space complexity with BFB, and the real time costs less than standard BFB due to shorter path length.

Algorithm 1: SBFB

Input: Markov chain $(\chi, \mathbb{P})$, initial state s_0, goal state g, step number k, sampling number N, minimum error η, sampling percentage $N^{\%}$
Output: simulation result $\hat{\pi}$

1 $\pi[0] := \{s_0 : 1\}$;
2 **for** $i := 1$ *to* k **do**
3 $\quad \pi[i] := \emptyset$ and default value is 0 ;
4 $\quad$ **foreach** $s \in \{s | \forall s \in \pi[i-1], \pi[i-1][s] > \eta\}$ **do**
5 $\quad\quad$ **foreach** $s' \in \{s' | \forall s' \in \chi, p_{ss'} > 0\}$ **do**
6 $\quad\quad\quad \pi[i][s'] := \pi[i][s'] + \pi[i-1][s] * p_{ss'}$;
7 $\quad\quad$ **end**
8 $\quad$ **end**
9 **end**
10 $m := len(\pi(k))$;
11 **foreach** $s_i, p(\pi_i)$ *in* $\pi[k]$ **do**
12 $\quad \hat{\pi}_i, \sigma_i :=$ BFB$(\chi, \mathbb{P}, s_i, N \cdot N^{\%}/m)$;
13 **end**
14 **foreach** $s_i, p(\pi_i)$ *in* $\pi[k]$ **do**
15 $\quad N_i := N \cdot (1 - N^{\%}) \cdot \frac{p(\pi_i)\cdot\sigma_i}{\sum_{k=1}^{m} p(\pi_k)\sigma_k}$;
16 $\quad \hat{\pi}_i, \sigma_i :=$ BFB$(\chi, \mathbb{P}, s_i, N_i)$;
17 **end**
18 $\hat{\pi} := \sum_{i=1}^{m} p(\pi_i) \cdot \hat{\pi}_i$;
19 return $\hat{\pi}$

B. Stratified Sampling on Distance-based BFB (SBFB-D)

The Stratified Sampling algorithm for Distance-based BFB has only a little difference. Distance-based BFB use distance information to change the original transform probability into more reasonable probability, and thus gain a better variance. So there would be two ways to insert the stratified sampling into this kind of BFB.

The first algorithm uses distance information to change the transform probability first, and then uses this changed probability to call SBFB algorithm.

The second is using Distance-based BFB on each of the subspace, get the variance of each subspace, allocate numbers of sampling to each subspace, and reuse the Distance-based BFB on the new sampling number to get the final result.

According to analysis and experiments, the second one is a little better. It has different transform probability on different subspace instead of using the same one, and this may achieve better variance with the sacrifice of a little time (changing probability according to distance information costs less time than path sampling).

The algorithm for SBFB-D of the second version is described in Algorithm 2. Compared to Algorithm 1, what has happened is only in line 12 and line 15, where line 12 saved the changed probability $\mathbb{Q}$ for each subspace, and line 15 used them again without another calculation.

Our Algorithm 2 needs k times of distance calculation time but less time due to shorter path length. Besides, its real time is also better than standard BFB.

Algorithm 2: SBFB-D

Input: Markov chain $(\chi, \mathbb{P})$, initial state s_0, goal state g, step number k, sampling number N, minimum error η, sampling percentage $N^{\%}$

Output: simulation result $\hat{\pi}$

1 $\pi[0] := \{s_0 : 1\}$;
2 **for** $i := 1$ *to* k **do**
3 $\quad \pi[i] := \emptyset$ and default value is 0 ;
4 $\quad$ **foreach** $s \in \{s | \forall s \in \pi[i-1], \pi[i-1][s] > \eta\}$ **do**
5 $\quad\quad$ **foreach** $s' \in \{s' | \forall s' \in \chi, p_{ss'} > 0\}$ **do**
6 $\quad\quad\quad \pi[i][s'] := \pi[i][s'] + \pi[i-1][s] * p_{ss'}$;
7 $\quad\quad$ **end**
8 $\quad$ **end**
9 **end**
10 $m := len(\pi(k))$;
11 **foreach** $s_i, p(\pi_i)$ *in* $\pi[k]$ **do**
12 $\quad \mathbb{Q}_i, \hat{\pi}_i, \sigma_i :=$ Distance-based BFB$(\chi, \mathbb{P}, s_i, N^{\%}/m)$;
13 **end**
14 **foreach** $s_i, p(\pi_i)$ *in* $\pi[k]$ **do**
15 $\quad N_i := N \cdot (1 - N^{\%}) \cdot \frac{p(\pi_i)\cdot\sigma_i}{\sum_{k=1}^{m} p(\pi_k)\sigma_k}$;
16 $\quad \hat{\pi}_i, \sigma_i :=$ BFB$(\chi, \mathbb{Q}_i, s_i, N_i)$;
17 **end**
18 $\hat{\pi} := \sum_{i=1}^{m} p(\pi_i) \cdot \hat{\pi}_i$;
19 return $\hat{\pi}$

C. Why it works

Our algorithms has three main advantages:

1) BRE property is met.
2) Lots of transitions in K^- are saved.
3) Cares more on more important paths.

First, according to BFB's property, BRE meets on BFB because its performance will not go down with a smaller ϵ. What we should do then is only to ensure ϵ's uncorrelation to the number of sampling and to ensure that the number of sampling is correlated to variance of subspaces. As the sample allocating approach described in Section III-A, although σ_i is correlated to ϵ, there would be the same relative values, and thus N_i is uncorrelated to variance of subspaces. In that case, BRE meets in SBFB and SBFB-D.

Second, most verification tasks would have many destroy and repair transitions at the same time, so transitions go out and back in K^-.

Third, since we have justify the sampling number from each state in K^*, our weighted mean of simulation result will need less sampling number in all than the ordinary mean one.

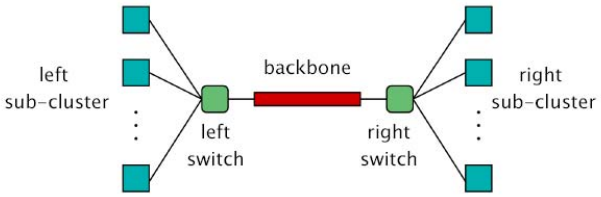

Fig. 1. Example of the case study of Section IV-A

TABLE II
SIMULATION RESULTS WITH DIFFERENT ϵ IN EXPERIMENT 1

ϵ	SBFB	SBFB-D	BFB
10^{-2}	$2.90 \cdot 10^{-3} \pm$ 0.06%	$2.88 \cdot 10^{-3} \pm$ 0.07%	$2.73 \cdot 10^{-3} \pm$ 10.4%
10^{-4}	$3.02 \cdot 10^{-7} \pm$ 0.07%	$2.99 \cdot 10^{-7} \pm$ 0.07%	$2.86 \cdot 10^{-7} \pm$ 11.4%
10^{-6}	$3.00 \cdot 10^{-10} \pm$ 0.07%	$3.12 \cdot 10^{-10} \pm$ 0.07%	$3.27 \cdot 10^{-10} \pm$ 10.9%
10^{-8}	$3.15 \cdot 10^{-13} \pm$ 0.07%	$3.26 \cdot 10^{-13} \pm$ 0.07%	$3.15 \cdot 10^{-13} \pm$ 11.1%

IV. Experimental Results

In this section, we have tried two experiments on Algorithm 1, Algorithm 2 and standard BFB. As our results, Algorithm 2 is better than Algorithm 1 when simulating HRMSs while Algorithm 1 is better when high redundancy exists. Besides, adding stratified sampling always gain a better variance.

A. Workstation Cluster

This case is based on a cluster of workstations, and the system comprises two sub-clusters with N workstations in each, connected in a star topology. The switches connecting each sub-cluster are joined by a central backbone. All components can break down and there is a single repair unit to service all components. We should verify that at least $3N/4$ workstations are operational and connected via switches and backbone.

First, we set $N = 12$, number of sampling $= 10000$, confidence $= 0.01$, and $\epsilon = 10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}$. Results are shown in Table II. The table shows the mean simulation result and their variance, and both these two kinds of our algorithm both have a much lower variance than standard BFB, and they all meet the BRE property.

TABLE III
SIMULATION RESULTS OF DIFFERENT REDUNDANCY IN EXPERIMENT 1

REDUNDANCY	SBFB	SBFB-D
4	$2.74 \cdot 10^{-4} \pm 0.67\%$	$2.96 \cdot 10^{-4} \pm 0.61\%$
16	$7.80 \cdot 10^{-4} \pm 0.78\%$	$7.46 \cdot 10^{-4} \pm 1.25\%$
64	$1.82 \cdot 10^{-5} \pm 1.06\%$	$1.79 \cdot 10^{-5} \pm 8.97\%$
128	$1.34 \cdot 10^{-5} \pm 1.94\%$	$9.04 \cdot 10^{-6} \pm 11.23\%$

TABLE IV
TIME SPENDING AND AVERAGE PATH LENGTH IN EXPERIMENT 1

	SBFB	SBFB-D	BFB
Average Path Length	4	4	5
Time Spending	12.30s	13.10s	26.44s

Second, $\epsilon = 1e-2$ and $N = 4, 16, 64, 128$ is set to analyze the influence of redundancy in Table III. Algorithm 1 is better when large redundancy is set, and its performance is always good.

Third, average path length and time spending is compared between different algorithms in Tabel IV. Time spending droped down for about 50% from standard BFB, mostly due to the reduction of average path length.

B. Distributed Database System

In This example, this system consists of 2 processors, 4 disk controllers and 6 disk clusters with 4 disk in each. As shown in Fig 2, this system would be seem as failure when one of these three states is met :

1) All of the processors are failed.
2) One of the groups of disk controllers are all failed.
3) One of the groups of disk clusters have more than one failed disks.

Besides, when a disk or a disk controller or a processor is failed, the repair program would repair it with rate 1. Here we set the failure rates $1/6000$ for disk and $1/2000$ for both controller and processor.

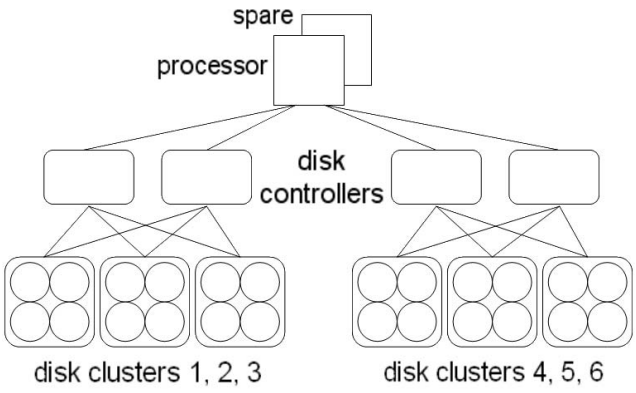

Fig. 2. Example of the case study of Section IV-B

TABLE V
SIMULATION RESULTS WITH DIFFERENT ϵ IN EXPERIMENT 2

ϵ	SBFB	SBFB-D	BFB
10^{-2}	$5.92 \cdot 10^{-3} \pm 3.934\%$	$5.87 \cdot 10^{-3} \pm 0.289\%$	$5.48 \cdot 10^{-3} \pm 29.69\%$
10^{-4}	$4.95 \cdot 10^{-5} \pm 4.083\%$	$4.96 \cdot 10^{-5} \pm 0.633\%$	$5.03 \cdot 10^{-5} \pm 10.85\%$
10^{-8}	$4.42 \cdot 10^{-7} \pm 2.134\%$	$4.41 \cdot 10^{-7} \pm 0.197\%$	$5.07 \cdot 10^{-7} \pm 11.14\%$

TABLE VI
TIME SPENDING AND AVERAGE PATH LENGTH IN EXPERIMENT 2

	SBFB	SBFB-D	BFB
Average Path Length	4.56	3.82	4.88
Time Spending	28.65s	17.40s	36.78s

We have tried number of sampling$= 10000$, confidence $= 0.01$ on this experiment, and results are shown in Table V and Table VI. As we can see, standard BFB have a better variance when added stratified sampling, and SBFB-D have a even better variance. It mainly dues to the failure path's better estimation on subspace's variance. Besides, Table VI also shows that SBFB-D can have a shorter average path length.

V. CONCLUSIONS

We have introduced a rare event simulation framework that is generally applicable to BFB and its variants. We have proved its efficiency both on theory and experiments, and it shows a great performance. From our experiments and some other papers like [14], Distance-based BFB always does bad on high component redundancy but has a good performance on other systems. In our experiments, our SBFB algorithm has a better performance than standard BFB, and SBFB-D has a better performance than Distance-based BFB, too. Besides, the gap between BFB and Distance-based BFB on HRMSs could be largely improved by stratified sampling. In short, systems with high component redundancy can use SBFB, and SBFB-D is better when simulating other systems.

There are several directions for future work. The simulation code has not been optimised for performance, so improving is a future work. There are still many other variance reduction techniques should be studied in more detail, and we could compare the performance of our method to a wider range of other IS techniques, e.g., the cross-entropy method.

REFERENCES

[1] M. Kwiatkowska, G. Norman, and D. Parker, "Prism 4.0: Verification of probabilistic real-time systems," in *International conference on computer aided verification*. Springer, 2011, pp. 585–591.
[2] J. Hammersley and D. Handscomb, "Monte carlo simulation," in *Monte Carlo methods*. Methuen & Co. Ltd. London, UK, 1964.
[3] E. E. Lewis and F. Böhm, "Monte carlo simulation of markov unreliability models," *Nuclear engineering and design*, vol. 77, no. 1, pp. 49–62, 1984.
[4] T. Zhuguo and E. E. Lewis, "Component dependency models in markov monte carlo simulation," *Reliability Engineering*, vol. 13, no. 1, pp. 45–61, 1985.

[5] A. E. Conway and A. Goyal, *Monte Carlo simulation of computer system availability/reliability models*. IBM Thomas J. Watson Research Division, 1986.
[6] A. Goyal, P. Heidelberger, and P. Shahabuddin, "Measure specific dynamic importance sampling for availability simulations," in *Proceedings of the 19th conference on Winter simulation*. ACM, 1987, pp. 351–357.
[7] P. Shahabuddin, V. F. Nicola, P. Heidelberger, A. Goyal, and P. W. Glynn, "Variance reduction in mean time to failure simulations," in *Proceedings of the 20th conference on Winter simulation*. ACM, 1988, pp. 491–499.
[8] A. Goyal, P. Shahabuddin, P. Heidelberger, V. F. Nicola, and P. W. Glynn, "A unified framework for simulating markovian models of highly dependable systems," *IEEE Transactions on Computers*, vol. 41, no. 1, pp. 36–51, 1992.
[9] P. Shahabuddin, "Simulation and analysis of highly reliable systems," 1990.
[10] J. A. Carrasco, "Failure distance-based simulation of repairable fault-tolerant systems," in *Computer performance evaluation: modelling techniques and tools: proceedings of the Fifth International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Torino, Italy, 13-15 February 1991*. Elsevier, 1992, pp. 351–365.
[11] ——, "Failure transition distance-based importance sampling schemes for thesimulation of repairable fault-tolerant computer systems," *IEEE Transactions on Reliability*, vol. 55, no. 2, pp. 207–236, 2006.
[12] S. Juneja and P. Shahabuddin, "Fast simulation of markov chains with small transition probabilities," *Management Science*, vol. 47, no. 4, pp. 547–562, 2001.
[13] D. Reijsbergen, P.-T. D. Boer, W. Scheinhardt, and S. Juneja, "Path-zva: General, efficient, and automated importance sampling for highly reliable markovian systems," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 28, no. 3, p. 22, 2018.
[14] D. E. Hocevar, M. R. Lightner, and T. N. Trick, "A study of variance reduction techniques for estimating circuit yields," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 2, no. 3, pp. 180–192, 1983.
[15] M. L. Stein, "An efficient method of sampling for statistical circuit design," *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 5, no. 1, pp. 23–29, 1986.
[16] D. P. Kroese, T. Taimre, and Z. I. Botev, *Handbook of monte carlo methods*. John Wiley & Sons, 2013, vol. 706.