Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-16 (2018) 85–90

Statistical Verification of PCTL Using Stratified Samples

Yu Wang [*,***] Nima Roohi [**] Matthew West [***]
Mahesh Viswanathan [****] Geir E. Dullerud [*,***]

[*] Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: {yuwang8,dullerud}@illinois.edu).

[**] Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: roohi2@cis.upenn.edu).

[***] Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: mwest@illinois.edu).

[****] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: vmahesh@illinois.edu).

Abstract: In this work, we propose a stratified sampling method to statistically check Probabilistic Computation Tree Logic (PCTL) formulas on discrete-time Markov chains with sequential probability ratio test. Distinct from previous statistical verification methods using independent Monte Carlo sampling, our algorithm uses stratified samples that are negatively correlated, thus give lower variance. The experiments demonstrate that the new algorithm uses a smaller number of samples for a given confidence level on several benchmark examples.

Keywords: Markov chains, Temporal logic, Variance reduction, PRISM, Sequential probability ratio test

1. INTRODUCTION

Statistical model checking has received considerable attention during the past decade (Younes, 2005b; Younes and Simmons, 2006; Sen et al., 2004, 2005a,b; Larsen and Legay, 2016; Clarke and Zuliani, 2011; Henriques et al., 2012), due to its scalability to large-scale real-world problems with complicated stochastic behavior (Roohi et al., 2017; Wang et al., 2015b,a, 2016; Zuliani et al., 2012). The general idea is to treat the problem of checking if a PCTL formula holds on a probabilistic system as a hypothesis testing problem. By drawing sample behaviors from the underlying probabilistic system and using proper statistical inference, statistical model checking determines whether the samples constitute a statistical witness to the satisfaction of the specification with high confidence.

Most statistical model checking algorithms previously proposed crucially rely on *independent* Monte Carlo sampling. Specifically, the underlying probabilistic system is "simulated" to generate a sample path and a new sample is drawn in the same manner in each round. Consequently, the samples are independent and identically distributed (i.i.d.).

The main thesis of this paper is that verification time can be significantly reduced if the statistical model checker draws *correlated* samples, as opposed to independent samples. To illustrate this, let us consider the core task of a statistical model checker, namely, to determine if the measure of executions satisfying a property ϕ is greater than some threshold p. Let us, for simplicity, assume that the truth of ϕ itself can be determined by a finite prefix of the execution. In such a situation, the model checker draws sample executions, determines how many of the executions satisfy ϕ, and uses this to estimate the measure of paths satisfying ϕ. Thus, each sample can be viewed as a 0/1-valued random variable X_i (which takes value 1 if the execution satisfies ϕ, and 0 otherwise), whose expectation is estimated by

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

One factor that plays an important role in determining how many samples are needed for the algorithm to be confident in its answer is the variance. Informally, the lower the variance of the estimate, the more likely the estimate is to be close to the actual mean, and therefore, the algorithm requires fewer samples. In general, the variance of the estimate is given by

$$\mathrm{Var}\left[\bar{X}\right] = \frac{1}{n^2}\sum_{i=1}^{n} \mathrm{Var}\left[X_i\right] + \frac{2}{n^2}\sum_{i=1}^{n}\sum_{j=i+1}^{n} \mathrm{Cov}\left[X_i, X_j\right].$$

If the samples are i.i.d., then the covariance is 0, and the variance is given by

$$\mathrm{Var}\left[\bar{X}\right] = \frac{1}{n}\mathrm{Var}\left[X\right].$$

⋆ This work is supported by NSF CPS Grant 1329991 and AFOSR Grant FA9550-15-1-0059.

2405-8963 © 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.08.015

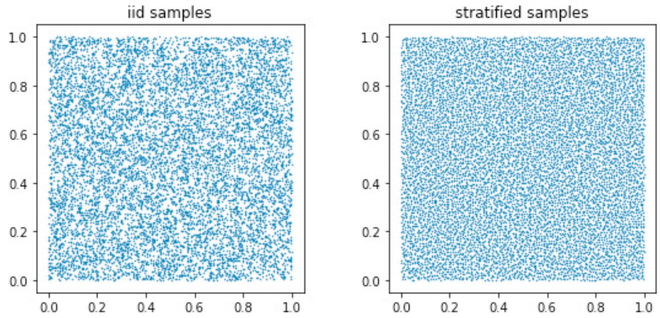


Fig. 1. Independent samples v.s. stratified samples on unit square

However, as can be seen from the above expression, the variance can be reduced if the samples are *negatively correlated*, i.e.,

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathrm{Cov}\left[X_i, X_j\right] \leq 0.$$

A common way to generate such negatively correlated samples with negligible additional computational cost is *stratified sampling*, which has been popular among the statistics community in improving the accuracy of statistical estimation (Liu, 2008; Hermanns et al., 2012; Maginnis et al., 2016). The general idea is to partition the sample space into different cells and draw one sample from each one of them. The stratified samples are repellent to each other — a sample occupying some cell forbids other samples entering the cell. Therefore, the stratified samples will be negatively correlated. For example, we can draw 10 000 stratified samples uniformly from the unit square $[0,1]^2$ by first partitioning the area into 100×100 small cells, each of size 0.01×0.01, and then draw exactly one sample from each cell. Figure 1 shows graphically that compared to 10 000 independent samples, 10 000 stratified samples are negatively correlated, hence distribute more evenly on $[0,1]^2$.

In this paper, we present a statistical model checking algorithm for checking finite horizon PCTL properties on discrete time Markov chains (DTMC) using stratified sampling. To ensure a lucid exposition of the main ideas, we only consider PCTL formulas of the form $\mathcal{P}_{\sim p}\phi$, where ϕ is a formula without probabilistic operator; in other words, ϕ's truth can be determined on single path. PCTL formulas in general form with nested probabilistic operators can be handled in the standard manner using the approach proposed in (Sen et al., 2004, 2005b,a). The main contribution of this paper is a sequential probability ratio test that works when samples are drawn using stratified sampling, which helps reduce the total number of samples (number of strata × number of blocks of stratified samples) needed for a statistical model checker to be confident in its answer.

The rest of the paper is organized as follows. The preliminaries on discrete-time Markov chains, probabilistic computational tree logic and stratified sampling are given in Section 2. In Section 3, we propose a sequential hypothesis testing algorithm using stratified sampling that gives the desired confidence level asymptotically. We have implemented this algorithm, and our preliminary experiments on several benchmarks are presented in Section 4. Finally, we conclude this work in Section 5.

2. PRELIMINARIES

We denote the set of natural numbers and real numbers by $\mathbb{N}, \mathbb{R}$. We take the convention that $0^0 = 1$. For $n \in \mathbb{N}$, let $[n] = \{1, 2, \ldots, n\}$. A permutation of $[n]$ is a bijection $\pi : [n] \to [n]$. For $p \in \mathbb{R}^n$, the i^{th} entry of p is denoted by p_i. For $M \in \mathbb{R}^{n \times m}$, the entry in the i^{th} row, j^{th} column of M is denoted by M_{ij}.

2.1 Markov Chains

Consider a discrete-time (homogeneous) Markov chain $\mathcal{M}$ of n numbered states with initial state $s \in [n]$ and transition probability matrix M, in which M_{ij} defines the transition probability from i to j. For any $j \in [n]$,

$$\sum_{i=1}^{n} M_{ij} = 1. \tag{1}$$

For a sample path $X = \{X(t)\}_{t \in \mathbb{N}} \subseteq [n]$ of the Markov chain, we can write

$$X(t+1) = f(X(t), E(t)), \quad t \in \mathbb{N} \tag{2}$$

where $E(t) \sim \mathbf{U}_{[0,1)}$. Generally, a discrete-time Markov chain $\mathcal{M}$ can be represented in (2) in multiple ways. In this work, we choose the following representation

$$f(i,e) = \begin{cases} 1, & \text{if } 0 \leq e < M_{i1} \\ j, & \text{if } \sum_{k=1}^{j-1} M_{ij} \leq e < \sum_{k=1}^{j} M_{ij}. \end{cases} \tag{3}$$

2.2 Stratified Sampling

As shown in (2), the Markov chain is driven by the random seed $E(t)$ uniformly sampled from the interval $\mathbf{U}_{[0,1]}$. Therefore, there is a bijection between the space of sample paths of the Markov chain $\mathcal{M}$ of length T and $[0,1]^T$. The stratified sampling algorithm generates m sample paths simultaneously. At each time t, the interval $[0,1)$ can be partitioned into m sub-intervals, namely $[0,1] = [0, \frac{1}{m}] \cup \ldots \cup [\frac{m-1}{m}, 1)$. Thus, a sample can be drawn from each sub-interval. To avoid correlation between steps, we generate a permutation π on $[n]$ uniformly at each time t, and then assign the sub-interval $[\frac{\pi(i)-1}{m}, \frac{\pi(i)}{m})$ to the i^{th} path. The random seeds of the m-stratified sample paths are repellent to each other in $[0,1]^T$, hence, due to the choice of $f(i,e)$ in (3), the m-stratified sample paths are repellent to each other in the space of sample paths. This is summarized by Definition 1 and Algorithm 1. Compared to i.i.d. samples, the additional computational cost for generating stratified samples is negligible.

Definition 1. $\{X_i\}_{i \in [m]}$ is called m-stratified samples if they are generated by Algorithm 1.

2.3 PCTL

Probabilistic computational tree logic (PCTL) is commonly used to express probabilistic properties of discrete-time Markov chains. In this paper, we only consider PCTL formulas in finite time horizon. The syntax and semantics of the logic is given below.

Algorithm 1 m-stratified sampling

```
Require: Number of strata $m$, number of steps $T$, and initial state $s$
$t = 0$
for $i \in [m]$ do
    $X_i(0) = s$
end for
for $t = 1, \dots, T-1$ do
    Take $\pi$ as a permutation of $[m]$
    for $i \in [m]$ do
        Take $E_i \sim \mathbf{U}_{[\frac{\pi(i)-1}{m}, \frac{\pi(i)}{m})}$
        $X_i(t+1) = f(X_i(t), E_i(t))$
    end for
end for
return $\{X_i\}_{i \in [m]}$
```

Definition 2. (Syntax). Let Ω be a given set of atomic propositions. A PCTL formula is generated recursively by

$$\phi ::= \Omega | \neg\phi | \phi \wedge \psi | \mathcal{P}_{\sim p}(\mathcal{X}\phi) | \mathcal{P}_{\sim p}(\phi \mathcal{U}_{\leq T} \psi), \tag{4}$$

where $\omega \in \Omega$ is an atomic proposition, $\sim \in \{<, >, \leq, \geq\}$, $p \in (0,1)$ is a probability threshold, $T \in \mathbb{N}$ is a time bound.

Remark 1. When $p = 0, 1$, the PCTL formula reduces to a CTL formula. In this work, we only consider $p \in (0,1)$. Other common temporal operators can be constructed by composing the temporal logic operators given in Definition 2.

Definition 3. (Semantics). Let $L : [n] \to 2^{\Omega}$ be a given labeling function where $[n]$ is the states of the Markov chain $\mathcal{M}$. A random trajectory starting from the state $s \in [n]$ is denoted by $s(0) = s, s(1), s(2), \dots$. The semantics of PCTL is defined recursively by

$$\begin{aligned}
&s \models \omega \text{ iff } \omega \in L(s)\\
&s \models \neg\phi \text{ iff } s \not\models \phi\\
&s \models \phi \wedge \psi \text{ iff } s \models \phi \text{ and } s \models \psi\\
&s \models P_{\sim p}(\mathcal{X}\phi) \text{ iff } \mathbb{P}[s(1) \models \phi] \sim p\\
&s \models \mathcal{P}_{\sim p}(\phi \mathcal{U}_{\leq T} \psi) \text{ iff}\\
&\quad \mathbb{P}[\exists t \leq T : s(0) \models \phi, \dots, s(t-1) \models \phi, s(t) \models \psi] \sim p
\end{aligned} \tag{5}$$

3. STATISTICAL VERIFICATION USING STRATIFIED SAMPLES

Now, we propose a sequential hypothesis testing algorithm using stratified sampling that gives the desired confidence level asymptotically. As mentioned in Section 1, we only consider PCTL formulas of the form $\mathcal{P}_{\sim p}\phi$, where ϕ is a formula without probabilistic operator. In other words, the correctness of ϕ can be determined for any trajectory X generated by the Markov chain $\mathcal{M}$. We define with a slight abuse of notation that

$$\phi(X) = \begin{cases} 1, & \text{if } X \text{ satisfies } \phi,\\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Consequently, checking $\mathcal{P}_{<p}\phi$ is equivalent to a composite hypothesis testing problem

$$\begin{aligned} &H_0 : \mathbb{P}[\phi(X)] < p,\\ &H_1 : \mathbb{P}[\phi(X)] \geq p. \end{aligned} \tag{7}$$

As with other literature in this area, we assume some a priori knowledge on the distance $|\mathbb{P}[\phi(X)] - p|$.

Assumption 1. Let $|\mathbb{P}[\phi(X)] - p| > \delta$ for some known indifference parameter $\delta > 0$. The interval $(\mathbb{P}[\phi(X)] - \delta, \mathbb{P}[\phi(X)] + \delta)$ is called the indifference Region.

Due to Assumption 1, the PCTL formulas $\mathcal{P}_{<p}\phi$ and $\mathcal{P}_{\leq p}\phi$, or $\mathcal{P}_{>p}\phi$ and $\mathcal{P}_{\geq p}\phi$ are equivalent. To be concrete, we consider $\mathcal{P}_{<p}\phi$ in the rest of this section; formulas in other forms can be deal with in similar ways.

With Assumption 1, the composite hypothesis testing problem can be simplified to a simple hypothesis testing problem by testing between the worst cases in the two hypothesis H_0 and H_1,

$$\begin{aligned} &H_0' : \mathbb{P}[\phi(X)] \leq p - \delta,\\ &H_1' : \mathbb{P}[\phi(X)] \geq p + \delta. \end{aligned} \tag{8}$$

When the sample paths $X_1, X_2, \dots$ are drawn independently, the hypothesis testing problem (8) can be solved efficiently with a sequential probability ratio test (SPRT) as shown in (Sen et al., 2004, 2005b,a). Specifically, for a confidence level of type I error

$$\alpha = \mathbb{P}[\text{choose } H_1' | \mathbb{P}[\phi(X)] = p - \delta] > 0, \tag{9}$$

and type II error

$$\beta = \mathbb{P}[\text{choose } H_0' | \mathbb{P}[\phi(X)] = p + \delta] > 0, \tag{10}$$

we consider the probability ratio

$$\Lambda(X^{(n)}) = \Pi_{i=1}^{n} \frac{(p+\delta)^{\phi(X_i)}(1-p-\delta)^{1-\phi(X_i)}}{(p-\delta)^{\phi(X_i)}(1-p+\delta)^{1-\phi(X_i)}}, \tag{11}$$

where $X^{(n)} = (X_1, \dots, X_n)$. H_0 is accepted if $\Lambda(X^{(n)}) > \frac{\beta}{1-\alpha}$; H_0 is accepted if $\Lambda(X^{(n)}) > \frac{1-\beta}{\alpha}$; otherwise, draw a new sample X_{n+1}.

3.1 Properties of Stratified Samples

To implement the SPRT on m-stratified samples $\{X_i\}_{i \in [m]}$, we consider the statistics

$$Y = \sum_{i=1}^{m} \phi(X_i)/m, \quad i = 1, 2, \dots. \tag{12}$$

By the generation of the stratified samples in Algorithm 1, we first note that

$$\mathbb{E}[Y] = \mathbb{E}\left[\sum_{i=1}^{m} \phi(X_i)/m\right] = \mathbb{E}[\phi(X_i)]. \tag{13}$$

In addition, we show below that for certain PCTL formulas ϕ, $\phi(X_i^1), \dots, \phi(X_i^m)$ can be generated negatively correlated, such that

$$\operatorname{Var}[Y] \leq \operatorname{Var}\left[\sum_{i=1}^{m} \phi(X_i)/m\right] = \operatorname{Var}[\phi(X_i)]/m. \tag{14}$$

By the syntax of PCTL, ϕ is either of the form $\mathcal{X}\psi$ or $\phi = \psi_1 \mathcal{U}_{\leq T} \psi_2$, where ψ_1 and ψ_2 are directly checkable on the states of the Markov chain $\mathcal{M}$. We denote the set of states where ψ holds by

$$V_{\psi} = \{s \in [n] | \psi \in L(s)\}. \tag{15}$$

Assumption 2. For a PCTL formula of the form $\phi = \psi_1 \mathcal{U}_{\leq T} \psi_2$, we assume (i) $V_{\psi_2} \subseteq V_{\psi_1}$; (ii) The states of the Markov chain $\mathcal{M}$ are numbered such that $V_{\psi_1} = [n_1]$ and $V_{\psi_2} = [n_2]$ where $n_1 \geq n_2$.

Theorem 1. With Assumption 2, let $\{X_i\}_{i \in [m]}$ be m-stratified samples from Markov chain $\mathcal{M}$ and ϕ be a probabilistic-operator-free PCTL formula with satisfaction probability p, then for any and $i \in [m]$,

(i) $\mathbb{E}\left[\sum_{i=1}^{m} \phi(X_i)/m\right] = \mathbb{P}\left[\phi(X)\right]$;
(ii) $\mathrm{Cov}\left[\phi(X_i), \phi(X_j)\right] \leq 0$ for $i \neq j$,

where X is a sample path drawn naively from the Markov chain $\mathcal{M}$.

Now, the hypothesis testing problem (8) can be converted to

$$\begin{aligned} H_0' &: \mathbb{E}\left[Y\right] = p - \delta, \\ H_1' &: \mathbb{E}\left[Y\right] = p + \delta. \end{aligned} \tag{16}$$

In addition, the mean of m-stratified samples within each block are more concentrated than the mean of m independent samples with the same mean,

$$\begin{aligned} \mathrm{Var}\left[Y_i\right] &= \frac{1}{m^2}\mathrm{Var}\left[\sum_{j=1}^{m} \phi(X_i^j)\right] \\ &= \frac{1}{m}\mathrm{Var}\left[\phi(X_i^j)\right] + \frac{1}{m}\sum_{k=1, k\neq j}^{m} \mathrm{Cov}\left[\phi(X_i^j), \phi(X_i^k)\right] \\ &\leq \frac{1}{m}\mathrm{Var}\left[\phi(X_i^j)\right]. \end{aligned} \tag{17}$$

Theorem 1 shows that compared to the mean m independent samples, the mean a group of m-stratified samples has the same mean, but smaller or at least equal variance. In addition, it shows that refining stratification always reduces the variance. Specifically, given an m-stratification, by refining each stratum into n strata, we can derive an mn-stratification. The new mn-stratified sampling algorithm will be no worse than the old m-stratified sampling algorithm.

Finally, we show that there is no loss of statistical information by considering Y_i given by (12) instead of $(X_i^1, \ldots, X_i^m)$.

Theorem 2. Let $\pi(x_1, \ldots, x_m)$ be the joint probability mass function of $\phi(X_1)$, $\ldots, \phi(X_m)$, then the value of p only depends on $\sum_{i=1}^{m} x_i$.

3.2 Sequential Probability Ratio Test

By Theorem 2, we only need to consider Y_i to solve (8). Now given $Y^{(n)} = (Y_1, \ldots, Y_n) \subseteq \{0, 1/m, \ldots, 1\}$, we can construct an SPRT algorithm similar to (11),

$$\Lambda'(Y^{(n)}) = \Pi_{i=1}^{n} \frac{\pi_{H_1}(Y^{(n)})}{\pi_{H_0}(Y^{(n)})}. \tag{18}$$

where π_{H_1} and π_{H_0} are the probability mass function of Y_i under hypothesis H_0 and H_1 respectively.

However, unlike the i.i.d. case in (11), the exact form of π_{H_1} and π_{H_0} is hard to derive. Therefore, for simplicity, we take an asymptotic approach via Central Limit Theorem. Let $\nu(Y^{(n)})$ be the empirical distribution given $Y^{(n)}$, then the Wald statistics converges to normal distribution $N(0,1)$ for large n

$$Z_n = \frac{\bar{Y}_i - \theta}{\sigma_i} \to N(0,1) \tag{19}$$

where $\theta = \mathbb{E}\left[Y\right]$ and

$$\bar{Y}_i = \frac{1}{i}\sum_{k=1}^{i} Y_k, \quad \sigma_i^2 = \frac{1}{i}\sum_{k=1}^{i}(Y_k - \bar{Y}_i)^2 \tag{20}$$

are the sample mean and sample variance respectively. Therefore, the probability ratio in (18) converges to

$$\Lambda'(Y^{(n)}) \to Ce^{-\frac{2(\bar{Y}_i - p)\delta}{\sigma_i^2}}, \quad n \to \infty, \tag{21}$$

for some normalizing constant C. In practice, this approximation is sufficiently accurate when the number of samples $n \geq 30$ and $\mathbb{E}\left[Y\right]$ is not close to the end points 0 and 1, since the converge of probability ratio (21) is faster. When $\mathbb{E}\left[Y\right]$ is close to 0 or 1, the distribution $\pi(y)$ of Y will become skew, and the convergence is slower (Agresti and Coull, 1998; Tony Cai, 2005). When the number of strata $m = 1$, the probability ratio (21) is equal (11) in large sample limit $n \to \infty$. Using (21), we can construct a sequential hypothesis testing algorithm (Algorithm 2).

Algorithm 2 SPRT using stratified samples

```
Require: Number of strata $m$, Probability threshold $p$, Indifference Parameter $\delta$, Confidence level $\alpha, \beta > 0$, Minimal number of samples $N$
$r \leftarrow 0$
$\nu \leftarrow \{0, \ldots, 0\} \in \mathbb{Z}^{m+1}$
while true do
    $r \leftarrow r + 1$
    Take $m$-stratified samples $\{X_{1,r}, \ldots, X_{m,r}\}$
    $Y_r \leftarrow \sum_{i=1}^{m} \phi(X_{i,r})$
    $\nu(Y_r) \leftarrow \nu(Y_r) + 1$
    if $r \geq N/m$ then
        $\mu_r \leftarrow \frac{\sum_{i=1}^{m+1} \frac{i-1}{m}\nu(i)}{\sum_{i=1}^{m+1}\nu(i)}$
        $\sigma_r^2 \leftarrow \left(\frac{\sum_{i=1}^{m+1}(\frac{i-1}{m})^2\nu(i)}{\sum_{i=1}^{m+1}\nu(i)} - \mu_r^2\right)/r$
        if $\mu_r - p < -\frac{\sigma_r^2}{2\delta}\ln(\frac{1-\alpha}{\beta})$ then
            Return $H_0$
        else if $\mu_r - p > \frac{\sigma_r^2}{2\delta}\ln(\frac{1-\beta}{\alpha})$ then
            Return $H_1$
        end if
    end if
end while
```

4. SIMULATION

The sequential probability ratio test algorithm using stratified samples (Algorithm 2) is implemented on a small scale toy example and several more complicated benchmarks from (PRISM). In all the simulations, we set the type I (9) error and type II error (10) to be 0.05, namely, the probability of the algorithm to make an error is always less than 5%. To guarantee sufficient accuracy of the probability ratio approximation (21), a minimal number of $N = 256$ samples is set for each run. The number of strata are taken to be $1, 2, 4, 8$. Accordingly, the minimal number of blocks are 256, 128, 64, 32; which are sufficient for large sample approximation (see Section 3.2) to hold.

Algorithm 2 is also compared with the sequential probability ratio test with independent samples proposed in (Sen et al., 2005a, 2004, 2005b), which is represented by SPRT in Table 2. The details of the simulation setups are given in Table 1.

Toy: A discrete-time Markov chain of three states uniquely labeled by $\{1, 2, 3\}$ with probability transition matrix

$$\begin{bmatrix} 0.583 & 0.333 & 0.084 \\ 0.417 & 0.417 & 0.166 \\ 0.278 & 0.444 & 0.278 \end{bmatrix}.$$

Check

$$\mathcal{P}_{>p}(s \neq 2)\mathbf{U}_{[0,10]}(s = 1),$$

namely, whether the probability that a path avoids state 2 and finally returns back to state 1 within 10 steps is greater than p. The estimated probability for $(s \neq 2)\mathbf{U}_{[0,10]}(s = 1)$ to hold is 0.794956586 by the average of 1 000 000 000 i.i.d. samples. Therefore, we set the experiment for the following three cases

$$(p, \delta) = \begin{cases} (0.794956586 - 0.010002, 0.01), \\ (0.794956586 - 0.005002, 0.005), \\ (0.794956586 - 0.001002, 0.001). \end{cases}$$

where δ is the indifference parameter serving as an input to Algorithm 2.

One Die: A fair die modeled by a discrete-time Markov chain of 13 states and 20 transitions proposed in (Knuth and Yao, 1976). Each state is labeled by only one of $s = 1, \ldots, s = 7$. Check

$$\mathcal{P}_{>p}\mathbf{F}_{[0,3]}(s > 6),$$

The estimated probability for $\mathcal{P}_{>p}\mathbf{F}_{[0,3]}(s > 7)$ to hold is 0.749987868 by the average of 1 000 000 000 i.i.d. samples. Therefore, we set the experiment for the following three cases

$$(p, \delta) = \begin{cases} (0.749987868 - 0.010002, 0.01), \\ (0.749987868 - 0.005002, 0.005), \\ (0.749987868 - 0.001002, 0.001). \end{cases}$$

Two Dice: The sum of two fair dice modeled by a discrete-time Markov chain of 45 states and 79 transitions proposed in (Knuth and Yao, 1976). Similar to One Die, the states are either transient with at most two transitions with equal probability or sinks. Each state is labeled by only one of $s = 1, \ldots, s = 34$. Check $\mathcal{P}_{>p}\mathbf{F}_{[0,4]}(s = 5)$. The estimated probability for $\mathbf{F}_{[0,4]}(s = 5)$ to hold is 0.249983470 by the average of 1 000 000 000 i.i.d. samples. Therefore, we set the experiment for the following three cases

$$(p, \delta) = \begin{cases} (0.249983470 - 0.010002, 0.01), \\ (0.249983470 - 0.005002, 0.005), \\ (0.249983470 - 0.001002, 0.001). \end{cases}$$

Election: Synchronous leader election protocol of 4 processors and 5 candidates proposed in (Itai and Rodeh, 1990), which is modeled by a discrete-time Markov chain of 1933 states and 2557 transitions. Check $\mathcal{P}_{>p}\mathbf{F}_{[0,1]}(600 < s < 630)$, where s is a numbering of the states. The estimated probability for $\mathbf{F}_{[0,1]}(600 < s < 630)$ to hold is 0.040002770 by the average of 1 000 000 000 i.i.d. samples. Therefore, we set the experiment for the following three cases

$$(p, \delta) = \begin{cases} (0.040002770 - 0.010002, 0.01), \\ (0.040002770 - 0.005002, 0.005), \\ (0.040002770 - 0.001002, 0.001). \end{cases}$$

Table 1. Summary of example models and testing formulas

Model	States	Transitions	Testing Formula
Toy	3	9	$\mathcal{P}_{>p}(s \neq 2)\mathbf{U}_{[0,10]}(s = 1)$
One Die	13	20	$\mathcal{P}_{>p}\mathbf{F}_{[0,3]}(s > 7)$
Two Dice	45	79	$\mathcal{P}_{>p}\mathbf{F}_{[0,4]}(s = 5)$
Election	1933	2557	$\mathcal{P}_{>p}\mathbf{F}_{[0,1]}(600 < s < 630)$

Table 2. Average number of samples needed and error probabilities for SPRT with independent samples and Algorithm 2 for different strata sizes on the examples.

Case	1		2		3	
Strata	Samples	Error	Samples	Error	Samples	Error
SPRT	2054.4	4.68%	8276.8	4.38%	185310.9	2.74%
1	2275.5	5.13%	8708.4	4.66%	186778.3	2.80%
2	2283.6	5.33%	8684.3	4.41%	187172.8	2.76%
4	2083.6	5.63%	8038.4	4.33%	174372.5	2.99%
8	1485	4.93%	5692.3	4.55%	123723.0	2.79%

(a) Toy

Case	1		2		3	
Strata	Samples	Error	Samples	Error	Samples	Error
SPRT	2403.7	4.65%	9470.1	4.40%	218546.6	2.90%
1	2638.6	4.64%	9956.6	5.14%	221318.4	3.25%
2	1759.7	3.94%	6772.1	4.01%	149272.3	3.08%
4	1898.1	4.48%	7474.6	4.32%	162201.5	2.91%
8	1803.7	4.56%	7027.1	4.27%	155766.0	2.88%

(b) One Die

Case	1		2		3	
Strata	Samples	Error	Samples	Error	Samples	Error
SPRT	2573.5	4.82%	10019.2	4.54%	221101.3	2.67%
1	2605.9	3.67%	9878.5	3.86%	220478.8	3.17%
2	1753.0	4.14%	6702.0	4.54%	148054.3	2.81%
4	1180.1	4.02%	4499.1	4.38%	98349.0	3.03%
8	994.7	4.41%	3843.9	4.21%	84064.6	3.00%

(c) Two Dice

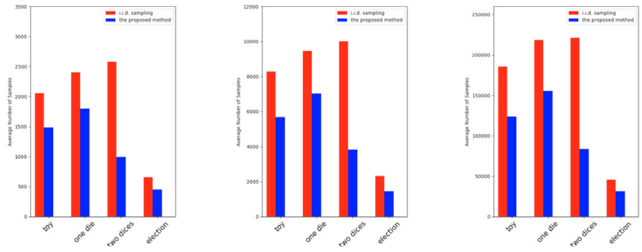
Case	1		2		3	
Strata	Samples	Error	Samples	Error	Samples	Error
SPRT	661.4	4.21%	2310.8	4.05%	45765.1	2.81%
1	586.4	1.33%	1976.1	1.91%	44506.4	2.20%
2	572.4	1.20%	1896.6	2.08%	42118.4	2.20%
4	535.8	1.57%	1758.4	2.22%	39154.0	2.43%
8	453.4	1.97%	1462.5	2.73%	31370.4	2.44%

(d) Election

The description of the simulation setups is summarized by Table 1. The simulation results for the above examples are shown in Table 2. The error probability and average sample size are derived by repeatedly running the algorithm for 10 000 to ensure statistical significance. The sample standard errors for the error probabilities and the average sample sizes are omitted in these tables for compactness.

The average sample size for Algorithm 2 for 1 stratum is approximately equal to the SPRT algorithm using independent samples. The former is always slightly larger than the latter, because there is a constraint on the minimal sample size. In all the cases, the actual type I error and type II error are controlled approximately below 0.05 with tolerable excess. These confirm that the large sample approximation used in Algorithm 2 is reasonable.

The reduction of sample size by stratification, as shown in Table 2a-2d, is visualized in Figure 2 below. The result shows that stratified sampling reduces the number of total samples (number of strata × number of blocks of stratified



(a) $\delta = 0.01$. (b) $\delta = 0.005$. (c) $\delta = 0.001$.

Fig. 2. Summary of reduction in average sample sizes for the toy, one die, two dice and election examples for three choices of indifference parameter δ.

samples), compared to independent sampling. Specifically, Algorithm 2 for 8 strata reduces the number of total samples by $30\% - 60\%$ in the four examples.

5. CONCLUSION

In this work, we propose a stratified sampling method to statistically check probabilistic computation tree logic formulas on discrete-time Markov chains with sequential probability ratio test. Compared to previous statistical verification methods using independent sampling, our algorithm uses stratified samples. They are negatively correlated, thus give a lower variance, while the additional computational cost for generating them is negligible. The experiments show that the latter uses $30\% - 60\%$ fewer samples (number of strata × number of blocks of stratified samples) than the former for a given confidence level on the benchmark examples.

ACKNOWLEDGEMENT

We would like to thank an anonymous reviewer for alerting us to a flaw in an earlier version of Theorem 1.

REFERENCES

PRISM - Case Studies. URL `http://www.prismmodelchecker.org/casestudies/index.php`.

Agresti, A. and Coull, B.A. (1998). Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician*, 52(2), 119–126.

Clarke, E.M. and Zuliani, P. (2011). Statistical model checking for cyber-physical systems. In *Automated Technology for Verification and Analysis*, 1–12. Springer, Berlin, Heidelberg.

Henriques, D., Martins, J.G., Zuliani, P., Platzer, A., and Clarke, E.M. (2012). Statistical model checking for Markov decision processes. In *2012 Ninth International Conference on Quantitative Evaluation of Systems*, 84–93.

Hermanns, H., Nielson, F., Jansen, David, N., and Zhang, L. (2012). Efficient CSL Model Checking Using Stratification. *Logical Methods in Computer Science*, 8, 2012.

Itai, A. and Rodeh, M. (1990). Symmetry breaking in distributed networks. *Inf. Comput.*, 88(1), 60–87.

Knuth, D. and Yao, A. (1976). Algorithms and complexity: New directions and recent results. Academic Press.

Larsen, K.G. and Legay, A. (2016). Statistical model checking: Past, present, and future. In *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques*, 3–15. Springer, Cham.

Liu, J. (2008). *Monte Carlo Strategies in Scientific Computing*. Springer.

Maginnis, P.A., West, M., and Dullerud, G.E. (2016). Variance-reduced simulation of lattice discrete-time Markov chains with applications in reaction networks. *Journal of Computational Physics*, 322, 400–414.

Roohi, N., Wang, Y., West, M., Dullerud, G., and Viswanathan, M. (2017). Statistical verification of the Toyota powertrain control verification benchmark. In *Proceedings of the 20th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '17, 65–70. ACM, New York, NY, USA.

Sen, K., Viswanathan, M., and Agha, G. (2005a). Vesta: A statistical model-checker and analyzer for probabilistic systems. In *Quantitative Evaluation of Systems, 2005. Second International Conference on the*, 251–252.

Sen, K., Viswanathan, M., and Agha, G. (2004). Statistical model checking of black-box probabilistic systems. In R. Alur and D.A. Peled (eds.), *Computer Aided Verification*, number 3114 in Lecture Notes in Computer Science, 202–215. Springer Berlin Heidelberg.

Sen, K., Viswanathan, M., and Agha, G. (2005b). On statistical model checking of stochastic systems. In K. Etessami and S.K. Rajamani (eds.), *Computer Aided Verification*, number 3576 in Lecture Notes in Computer Science, 266–280. Springer Berlin Heidelberg.

Tony Cai, T. (2005). One-sided confidence intervals in discrete distributions. *Journal of Statistical Planning and Inference*, 131(1), 63–88.

Wang, Y., Roohi, N., West, M., Viswanathan, M., and Dullerud, G.E. (2016). Verifying continuous-time stochastic hybrid systems via Mori-Zwanzig model reduction. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, 3012–3017.

Wang, Y., Roohi, N., West, M., Viswanathan, M., and Dullerud, G.E. (2015a). A Mori-Zwanzig and MITL based approach to statistical verification of continuous-time dynamical systems. *IFAC-PapersOnLine*, 48(27), 267–273.

Wang, Y., Roohi, N., West, M., Viswanathan, M., and Dullerud, G.E. (2015b). Statistical verification of dynamical systems using set oriented methods. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, HSCC '15, 169–178. ACM, New York, NY, USA.

Younes, H.L.S. (2005a). Probabilistic verification for "black-box" systems. In K. Etessami and S.K. Rajamani (eds.), *Computer Aided Verification*, number 3576 in Lecture Notes in Computer Science, 253–265. Springer Berlin Heidelberg.

Younes, H.L.S. (2005b). Ymer: A statistical model checker. In K. Etessami and S.K. Rajamani (eds.), *Computer Aided Verification*, number 3576 in Lecture Notes in Computer Science, 429–433. Springer Berlin Heidelberg.

Younes, H.L.S. and Simmons, R.G. (2006). Statistical probabilistic model checking with a focus on time-bounded properties. *Information and Computation*, 204(9), 1368–1409.

Zuliani, P., Baier, C., and Clarke, E.M. (2012). Rare-event verification for stochastic hybrid systems. In *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '12, 217–226. ACM, New York, NY, USA.