RESEARCH PAPER
Contrib. Sci. & Tech Eng, 2026, 3(1)
Homepage:https://cste.journals.umz.ac.ir/
DOI: 10.22080/cste.2025.28924.1025

Upper Boundof Policies for Statistical Model Checking of Markov Decision Processes

Mohammad Sadegh Mohagheghi [1]*

[1] *Assistant Professor,Department of Computer science, Vali-e-Asr University of Rafsanjan,Rafsanjan, Iran.*

Article Info
Received 03 April 2025
Accepted 29 September 2025
Available online 12 February 2026

Keywords:
Formal Verification;
Markov Decision Process;
Statistical Model Checking;
Confidence Interval.

Abstract:

State explosion remains a fundamental challenge in model checking. Statistical model checking (SMC) offers a promising approach to mitigate this issue. Still, its application to Markov decision processes (MDPs) is hindered by the computational difficulty of resolving nondeterminism with near-optimal policies. Existing methods for policy selection within SMC often suffer from limited memory efficiency or fail to guarantee the specified confidence level for result accuracy. To address this, we introduce a novel statistical criterion for policy evaluation and propose an efficient method for determining an upper bound on the number of policies required to achieve a desired confidence level in the computed reachability probabilities. Our method leverages the mean and standard deviation of reachability probabilities obtained from a set of randomly sampled policies to derive this bound. Experimental results validate the effectiveness of our approach and demonstrate that it provides more reliable results in most cases.

***Corresponding Author:** mohagheghi@vru.ac.ir

Supplementary information: Supplementary information for this article is available at https://cste.journals.umz.ac.ir/

Please cite this paper as: Mohagheghi, M. (2025). Upper bound of Policies for Statistical Model Checking of Markov Decision Processes. Contributions of Science and Technology for Engineering, 3(1), 31-40. doi:10.22080/cste.2025.28924.1025.

1. Introduction

Formal verification employs mathematically-driven methodologies to evaluate critical system properties in hardware and software. Among formal verification techniques, model checking automatically verifies whether a system model satisfies specified requirements [1, 2]. Systems are typically modeled as transition systems, whereas temporal logic formalisms specifybehavioral properties. Model checkers systematically examine system models to either validate compliance with requirements or generate counterexamples for violated properties. For stochastic systems, probabilistic extensions of transition systems, such as discrete-time Markov chains (DTMCs) and Markov decision processes (MDPs), are used. The Probabilistic Computational Tree Logic (PCTL) framework supports the specification of both qualitative and quantitative system characteristics. These analyses often focus on calculating reachability probabilities, defined as the probability of reaching a goal state from an initial state. In MDP analysis, determining extremal (minimum or maximum) probabilities becomes essential. Probabilistic model checking employs iterative graph-based algorithms and numerical methods to compute these metrics [3, 4].

A well-known challenge of model checking across all variants is the state explosion problem, i.e., exponential growth in a model's state space with the number of system components. A vast number of solutions have beenproposed in the past three decades to alleviate this challenge. Most of them are also customized for stochastic systems [2, 3]. Statistical model checking (SMC) as a simulation-based approach with small (usually O(1)) space complexity has been proposed and extended in recent years [5]. It simulates a set of traces to verify a given property. For the case of quantitative properties, SMC is expected to approximate the needed values by a given level of statistical confidence [6]. Consideringa bound for tolerating errors in computed values and δ for the probability of false precision (abusing the desired bound of errors), SMC uses enough Monte Carlo simulations to guarantee that the computed value is precise with the proposed confidence interval (CI) [6-9].

For MDPs models, SMC needs to resolve non-deterministic choices. In model checking of MDPs, the notion of a policy, defined as a mapping from states to actions, is a standard approach to resolving these choices [1, 3]. For each policy, a Discrete time Markov chain (DTMC) is induced and the problem is reduced to analyzing the behavior of the induced DTMC. In exhaustive probabilistic model checking, iterative methods (like policy iteration [3]) are used to approximate the optimal policy that maximizes (or minimizes) a reachability probability. However, an explicit representation of policies must store the optimal

ISSN3060-6578

action for each state, which contradicts SMC'sO(1) space complexity. Several techniques have been proposed to approximate optimal policies for MDPs with SMC. These techniques primarily rely on considering a large set of policies or employing elimination techniques to identify a near-optimal policy [6, 10-14]. While these techniques yield promising results for some classes of case study models, they cannot be generalized to all classes of MDPs, and their main challenge is that they do not guarantee the precision of the computed value at the desired confidence level. An example of a complex case is the consensus (coin) MDP model, which represents a distributed agreement protocol in which processes repeatedly vote and flip coins to reach consensus. The objective is to maximize the probability of reaching a unanimous agreement within a deadline [15]. Finding the optimal policy is challenging due to state-space explosion: the model's state space grows exponentially with the number of processes and rounds, and nondeterminism requires evaluating all possible schedulers that resolve process decisions at each step. Additionally, the probabilistic branching from coin flips multiplies computational complexity, making exact dynamic planning intractable for non-trivial instances. These factors collectively necessitate heuristic methods or approximations instead of exact solutions.

Focusing on unbounded extremal reachability probabilities, this paper develops new techniques to address the aforementioned SMC problems for MDPS. The possibility of applying these techniques to the class of bounded reachability probabilities remains for future work.To do so, we define a criterion for optimality of a given policy. It is based on comparing the average of computed values for the given policy with the average of exact values. For a near-optimal policy, the first value converges to the second one. Based on this criterion, we provide a CI for the relative distance of computed values to their exact ones and we can analyze the precision of the current techniques for approximating optimal policies. We then develop a method to determine an upper bound for the minimum number of policies that should be examined to find a ”near-optimal” one, i.e., the one that is guaranteed to be in a confidence interval with the proposed for errors and $1-\delta$ probability of confidence. To the best of our knowledge, no previous work has proposed such an upper bound for the number of needed policies. Our proposed method is based on the central limit theorem [16, 17].

1.1. Related Works

The first work on resolving non-deterministic choices in MDPs is the Kearns sparse sampling algorithm for discounted Markov reward models [18]. It is based on unfolding a model to a limited depth to approximate the optimal action for each state. For reachability properties of MDPs, a SMC algorithm is presented by Henriques et al. [11] that determines near-optimal policies with a specified probability of satisfying the underlying LTL property. For the PCTL class of properties, a SMC algorithm is developed by Ashok et al. [19] that uses bounded real-time dynamic programming to approximate the optimal policy with a given level of precision. These two algorithms are implemented in the PRISM model checker. While both works employ machine learning to converge to optimal policies, their efficiency depends on the structure of the underlying MDP. Both approaches require storing selected actions for the explored states during simulations [7, 8]. In the worst case, their space complexity is O(n), which violates the memory-independence property of SMC methods. Hence, they are typically applicable to moderate-sized models. Our proposed approach, however, requires a limited amount of memory regardless of model size and can be used with large models. On the fly algorithms are proposed in literature [7, 14, 20] that focus on the presented PCTL property to remove spurious non-deterministic choices of a model.Again, this class of techniques depends on the structure of the model and cannot resolve all non-deterministic choices in any MDP. Compared with these techniques, our proposed technique is independent of the model structure and applies to all MDP cases.

To address these challenges, policy sampling methods are proposed in literature [6, 8, 18] that use a four-bit integer and a pseudorandom number generator to generate a set of random memoryless policies. These methods (referred to as lightweight sampling methods for SMC) are based on considering a large set of random policies and approximating reachability probabilities by applying standard SMC to the induced DTMCs. They are developed as the plasma-lab model checker and can utilize multi-core processing to reduce runtime. These sampling methods rely on the assumption that with a high probability, a near-optimal policy can be found in this set. Based on this assumption, they propose a method to determine the number of samples required to approach the optimal state values and to assess the satisfaction of the proposed PCTL property. However, this assumption is not always correct, and there is no guarantee of the soundness of the approximated values at a specified confidence level. The main benefit of our approach is that it provides an upperbound on the number of policy samples required to satisfy the conditions for aconfidence interval. In addition to studies by Henriques et al. [11], Budde et al. [13], and Ashok et al. [19], several recent works have proposed using machine learning (ML) methods to improve the performance of SMC sampling for MDPs. ML is used by Rataj and Woźna-Szcześniak [21] for extrapolating the optimal policy of a given large MDP model. Deep neural networks have been used by Gros et al. [22] as black box determiners for MDP choices. The precision of the computed values in these two works is evaluated using several experimental results, but no formal approximation for the soundness of their proposed results has been proposed. Although their space complexity is independent of model size, it is unclear how they can be applied to any given large model. Moreover, another challenge of ML-based methods is that they may converge to local optima, and there is no known way to avoid this in statistical model checking. Our approach considers a large set of policies and identifies at least one near-optimal policy that satisfies the given CI conditions. An extension of smart sampling for SMC of rare events in MDPs is proposed by Budde et al. [12]. It is developed as a mode-based model checker [23], relies on the same assumptions as Legay et al.

[8], and proposes confidence intervals for simulation precision.

The SMC has been used in recent years to cover other classes of properties. A PAC statistical model-checking approach for mean-payoff games has been proposed by Agarwal et al. [24]. A PAC Learning Algorithm has been proposed by Perez et al. [25] for LTL and Omega-Regular Objectives over MDP models. Decision trees, as a class of machine learning techniques, are used by Azeem et al. [26] to yield more reliable results, whereas SMC is applied to MDP models.

2. Preliminaries

We review important concepts about Markov decision processes and statistical model checking. For a countable set R we use $|R|$ as its size, i.e., the number of its members. For a random variable z we use $\phi(z)$ as its standard normal distribution.

2.1. Markov Decision Processes

Definition 1. A Markov Decision Process (MDP) is a tuple$M = (S, s_0, Act, P, G)$ where S is a finite set of states, $s_0 \in S$ is the initial state, Act is a finite set of actions where for every state $s \in S$, $Act(s)$ denotes the non-empty set of enabled actions for s; $P: S \times Act \times S \to [0,1]$ is the transition probability function such that for each state $s \in S$ and each enabled action $\alpha \in Act$ we have: $\sum_{s' \in S} P(s, \alpha, s') \in \{0,1\}$. $G \subset S$ is a set of target states. In this definition, transition is a triple (s, α, s') for which$s, s' \in S$ and $\alpha \in Act$ and $P(s, \alpha, s') > 0$. For any state $s \in S$and an enabled action $\alpha \in Act(s)$ we define $Post(s) = \{t \in S | P(s, \alpha, t) > 0\}$ as the set of $\alpha -$successors of s. A path in an MDP, is a non-empty (finite or infinite) sequence of states and actions of the form $\omega = s_0\alpha_0 \to s_1\alpha_1 \to s_2\alpha_2 \to$ ... where $P(s_i, \alpha_i, s_{i+1}) > 0$ for each $i \geq 0$. We use $\omega(i)$ to denote the $(i+1)th$ state inthe path ω. $Path_s$ denotes the set of all paths starting in state s and $FPath_s$ is the set of all finite paths. A discrete-time Markov chain (DTMC) is an MDP in which every state has exactly one enabled action [1]. The successor state of each state of an MDP is determined in two steps. For any state $s \in S$, the first step selects one of the enabled actions $Act(s)$ non-deterministically. The second step, selects the next state randomly using the probability distribution $P(s, \alpha)$.

In the probabilistic model checking, a set of properties are specified in PCTL or probabilistic LTL. In this paper, we focus on the extremal unbounded reachability probabilities, defined as the maximum or minimum probability of reaching a target state G [1, 3]. To analyze the probabilistic behaviour of an MDP M, the notion of policy (also called adversary or scheduler) is used to resolve its non-deterministic choices. In this paper, we consider only deterministic, memoryless policies, which are sufficient for computing the extremal reachability probabilities [3, 27, 28]. A (deterministic and memory-less) policy for M is a function $\pi: S \to Act$ that for every state $s \in S$ selects an action $\alpha \in Act(s)$. We use Pol_M for the set of all policies of M. Each policy $\pi \in Pol_M$ induces a DTMC M^π by disregarding non-selected actions. Verifying reachability probabilities for a given MDP is reduced to finding an optimal policy π^* that optimizes reachability probabilities of all states. [1-3]. For unbounded reachability probabilities, memory-less policies are enough to compute the optimal values. For bounded reachability probabilities, however, memory-based policies are needed [3]. Iterative numerical methods, such as value iteration, policy iteration, and interval iteration [3, 4] are used in practice to approximate extremal reachability probabilities. For each state $s_i \in S$ an iterative method uses a variable x_i as an approximation of its maximal reachability probability. We use x_i^* as the computed (or approximated) optimal value for s_i and define the mean of optimal values as $MOV = \frac{1}{n}\sum_{1 \leq i \leq n} x_i^*$. We may use MOV^M to denote MOV for a given model M if we have several models. Iterative methods for probabilistic model checking and their soundness are explained by Baier and Katoen [1], Baier et al. [2], and Hartmanns and Kaminski [4].

2.2. Statistical Model Checking (SMC)

Instead of applying exhaustive search on a given model, SMC generates a set of sample runs to simulate the behaviour of a model. The model checker verifies each run to determine whether it satisfies the given property or not. After simulation, the sample statistics are used to estimate the error rate at a specified confidence level, satisfying the given property [9, 12, 29]. To provide a guarantee for the precision of the approximated values, a statistical model checker should determine the number of simulation runs based on the given minimum error rate ϵ and confidence level δ. Based on these parameters, a confidence interval (CI) is proposed for the correctness of the approximated values. The standard way in SMC for DTMC models is to determine M as the number of simulation runs such that $Pr(|x^N{}_0 - x_0| \leq \epsilon) \geq 1 - \delta$ where ${x_0}^N$ is the approximated value for s_0 after running N iterations. Most model checkers follow Hoeffding's Inequality as: $M \geq \frac{ln(\frac{2}{\alpha})}{2\epsilon^2}$. It is used when a model checker supports CI for verification of probabilistic systems [20, 27, 30]. In this paper we rely on CI for reachability probabilities of DTMCs, i.e., for any given MDP M, policy π , thresholds ϵ and δ for errors and confidence level, we follow the standard approaches to determine the number n of required simulations to guarantee the precision of values at the desired level [7, 8, 12, 31, 32].

2.3. Central Limit Theorem

The Central Limit Theorem (CLT) states that when independent random variables are summed or averaged, their distribution tends toward a normal distribution (Gaussian) as the sample size increases, regardless of the original population's distribution. Essential assumptions that need to apply CLT are as follows:

1. Independence: Observations must be independent, e.g., random sampling with replacement or from an infinite population.

2. Finite Mean and Variance: The population must have a finite mean (μ) and finite variance (σ 2≠∞).

3. Sample Size (n) must be sufficiently large (n > 50 in most cases).

3. The Proposed Method for an Upper Bound for the Number of Policy Samples

A main drawback of the smart sampling method and its extensions for SMC of MDPs is that their computational precision relies on the likelihood of finding a near-optimal policy among a set of randomly selected policies [12]. There are MDP models for which a main part of non-deterministic choices are spurious, or the impact of choosing an optimal action over the other ones is negligible for most states [7, 30]. In such cases, smart sampling can find a near-optimal policy with high probability. However, there are cases for which the number of policies increases exponentially in the number of states, while infinitely small parts of them are near-optimal.

Example 1 .Consider the MDP of Figure 1 with n states where s_n is the goal state and s_{n-1} a trap state and any other state have two actions. There are 2^{n-2} policies for this MDP, of which only one is optimal. For any policy with k non-optimal actions, the probability of reaching the goal from s_0 is $\frac{1}{2^k}$. Consider a case in which we need to find a policy that is at least 50% near-optimal, meaning that at most one non-optimal action is acceptable. There are $n - 1$ policies from 2^{n-2}ones that satisfy this condition. As a result, with a probability of $\frac{n-1}{2^n}$, A randomly selected policy is at least 50% optimal. For n = 50, this probability is $\frac{49}{2^{48}} = 1.7E-13$ that means with a probability of $1 - 1.74E - 13$. A randomly selected policy is less than 50% optimal. Having m different randomly selected policies, the probability of 50% near-optimality is $1 - (1 - 1.74E - 13)^m$. For this example, one should consider $\frac{log\ (0.1)}{log\ (1-1.74E-13)} = 1.32E + 13$ policies to be 90% confident to have at least one 50% near-optimal policy.

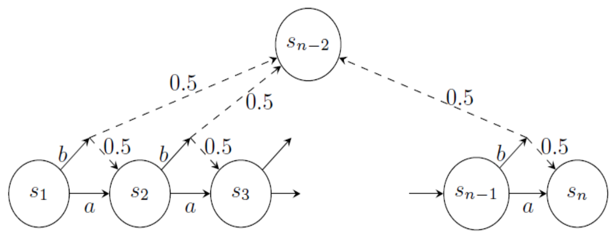

Figure 1. An MDP example with 2^{n-2} Possible Policies

3.1. Policy Evaluation

To assess how well a policy approximates reachability probabilities and how well it can be used in SMC, we define a criterion for policy optimality. For each state $s_i \in S$ we use x_i^* for its extremal probability of reaching a target state. For a policy π we use x_i^π for the probability of reaching a target state under π. We use $Abs\_err(x_i^\pi)$ and $Rel\_err(x_i^\pi)$ for the absolute and relative errors in the computed reachability values under π: $Abs\_err(\pi) = |x_i^* - x_i^\pi|$ and $Rel\_err = \frac{|x_i^* - x_i^\pi|}{max\ \{x_i^*, x_i^\pi\}}$. In this section, we define a criterion for the fitness of a given policy by comparing the computed reachability probabilities of its induced DTMC with their exact values. A standard criterion for computation errors is the mean square error of computed values [28]. However, due to technical constraints, we consider mean errors.

Definition 1. (Mean absolute and mean relative errors) For an MDP M and a given policy πwe use $MAE(\pi)$ and $MRE(\pi)$ for mean absolute errors and mean relative errors of computed reachability probabilities under π and define them as:

$$MAE(\pi) = \frac{1}{n}\sum_{i=1}^{n} Abs\_err(x_i^\pi) = \frac{1}{n}\sum_{i=1}^{n} |x_i^\pi - x_i^*| \quad (1)$$

and

$$MRE(\pi) = \frac{1}{n}\sum_{i=1}^{n} Rel\_err(x_i^\pi) = \frac{1}{n}\sum_{i=1}^{n} \frac{|x_i^\pi - x_i^*|}{max\ \{x_i^\pi, x_i^*\}} \quad (2)$$

For maximal reachability probabilities $x_i^* \geq x_i^\pi$ holds for each $s_i \in S$. In this case, for Equation 1 we have $MAE(\pi) = \frac{1}{n}\sum_{i=1}^{n} x_i^* - \frac{1}{n}\sum_{i=1}^{n} x_i^\pi$. On the other hand, for minimal reachability probabilities $x_i^* \leq x_i^\pi$ holds for each state and we have $MAE(\pi) = \frac{1}{n}\sum_{i=1}^{n} x_i^\pi - \frac{1}{n}\sum_{i=1}^{n} x_i^*$. Similar cases hold for mean average errors. For maximal reachability probabilities $MRE(\pi) = \frac{1}{n}\sum_{x_i^* \neq 0} \frac{x_i^* - x_i^\pi}{x_i^*}$ and minimal reachability probabilities $MRE(\pi) = \frac{1}{n}\sum_{x_i^* \neq 0} \frac{x_i^\pi - x_i^*}{x_i^\pi}$.

Mean relative error provides a more precise estimate of the likelihood of policy optimality. However, this method requires knowing the exact values x_i, which are not always available. Instead, mean absolute error (MAV) can be computed using the mean of exact values (MOV). MOV can be approximated without precisely estimating each x_i . In the next section, we propose an approach for approximating MOV. The ability to approximate MOV for MAV computation is the primary reason for using mean absolute error in Equation1 instead of mean squared error.

Using different random policies π, one can consider x_i^π as a random variable. In SMC, we are interested in controlling the precision of the approximated value for s_0. In general, there is not any assumption about the probability distribution of x_i^π over a set of random policies [16]. For SMC of MDPs, we focus on Equations 1 and 2 to control the precision of the approximated value in the induced DTMCs. In general, for a random policy π with $MRE(\pi) = \delta$ we expect $Rel\_err(x_0^\pi)$ to be around δ unless we have some more information about the probability distribution of (x_0^π). In the case of lack of knowledge about this distribution, we use the following lemma to relate $Rell\_err(x_0^\pi)$ to $MRE(\pi)$. In the remainder of this paper, we rely on this lemma and focus on $MRE(\pi)$.

Lemma 1. For a given policy $\pi \in POL_M$ let $MRE(\pi) = \delta$. For an error rate ϵ and any state $s_i \in S$, in the worst case (over all possible policies) we have:

$$Pr\ (Rell\_err(x_i^\pi) \geq \epsilon) = \frac{\delta}{\epsilon} \quad (3)$$

which means with probability $1-\frac{\delta}{\epsilon}$, for any state s_i we have x_i^{π} within $\mp\epsilon$ of x_i.

Proof: Suppose in the worst case the probability that $Rel\_err(x_i^{\pi}) \geq \epsilon$ is β. The worst case happens when for $100 \times \beta$ percent of states $s \in S$, $Rel\_err(x^{\pi})$ is equal to ϵ and for the other states $Rel\_err(x^{\pi})$ is equal to 0. On the other side, the average of this value should be δ, which means $\epsilon \times \beta + 0 \times (1-\beta) = \delta$. As a result, we have $\beta = \frac{\delta}{\epsilon}$.

Example 2. Consider for a selected policy π we have $MRE(\pi) = 0.01$ and let $\epsilon = .1$ as the maximum error rate. Using lemma 1 we have $Pr(Rel\_err(x_0^{\pi}) \geq 0.1) \leq 0.1$ that means with the probability of .9 (as confidence level) the relative error of the computed value for the initial state is 0.1. Note that we may conclude more precise CI if we have some information about probability distribution of x_0^{π} over different policies π. Note that lemma 1 also holds if one considers absolute errors, i.e., using $MAE(\pi)$ and $Abs\_err(x_i^{\pi})$ in the lemma. Having ϵ and δ as the CI parameters, one can use Lemma 1 to compute the required precision of the total computed values.

3.2. Upper-bound for Number of Policies

A key step in SMC is evaluating the precision of the approximation. For MDPs, it should also assess the precision of the selected policies and determine how many policies are needed to obtain a near-optimal policy that satisfies the conditions of the proposed confidence interval. Considering $Rand\_Pols$ as a set of randomly selected deterministic policies, we develop a method to compute an upper boundfor the size of $Rand\_Pols$ to have at least one near-optimal policy,i.e., for a given probability δ of confidence and an error bound ϵ and for the best policy $\pi_{best}$of $Rand\_Pols$ we have $Pr\ (MRE(\pi_{best}) \geq \epsilon)$. To the best of our knowledge, no previous work has proposed such a bound on the number of random policies for satisfying the conditions of precision except some memory-dependent works [11, 19]. Note that the smart sampling method [6] compares the true probability of the best candidate policy found with the estimated probability of the best candidate policy by considering a sample set of policies.

Using Pol_M as the set of all possible random policies of the given MDP M, one can consider MRE as a random variable over this set of policies. Having the probability distribution of this random variable, it is possible to determine the number of policies that satisfy the conditions of CI. In general, the structure of a model affects this probability distribution. For a large model with a huge number of states, we expect that with a high probability, for each pair of states (s_i, s_j). The impact of their value on one another is negligible. As a result, we can use the central limit theorem (CLT) and consider a normal distribution for MRE. More precisely, the second and third assumptions for CLT (as proposed in 2.3) hold for large MDP models: the state space is large enough and both the mean and variance of values are always finite because each state value is less than or equal to 1. For the first assumption, if there are a limited number of Dirac transitions, we expect that the value of each state depends on the values of a small subset of states, and, with high probability, the values of any two states are independent. The normal distribution is also used as default in *modest* for the mean of state values [16]. Although in general, considering normal distribution for MRE may not be correct, our experiments show that it holds for most case studies. Considering a normal distribution for MRE (or MAE), we apply the following lemma to compute an upper bound for the number of policies to find a near-optimal one.

Lemma 2. Suppose MRE is a random variable with normal distribution with μ and σ as its mean and standard deviation. Let $N = |Rand\_Pols|$ be the size of a sample set of policies. For a given probability δ and error bound ϵ, there exists a policy $\pi_{best} \in Rand\_Pols$ where $Pr(MRE(\pi_{best}) \geq \epsilon) \leq \delta$ if:

$$N \geq \left[\frac{log\ (\delta)}{log\ (1-\Phi\left(\frac{\epsilon-\mu}{\delta}\right))}\right] \quad (4)$$

Proof. Based on the assumptions, for a random policy $\pi \in Pol_M$ the probability that its MRE is less than ϵ is computed as:

$$PR(MRE(\pi) < \epsilon) = Pr\left(\frac{MRE(\pi)-\mu}{\sigma} < \frac{\epsilon-\mu}{\sigma}\right) = \Phi\left(\frac{\epsilon-\mu}{\sigma}\right) \quad (5)$$

As a result, $Pr(MRE(\pi) \geq \epsilon) = 1 - \Phi(\frac{\epsilon-\mu}{\sigma})$. As for any two policies π_1, π_2 the reachability probability of state in the induced DTMCs are independent, we consider $MRE(\pi_1)$ and $MRE(\pi_2)$ as two independent events. Hence, for a $Rand\_Pols$ set with N policies, the probability that for all policies $\pi \in Rand\_Pols$ we have $Pr(MRE(\pi) \geq \epsilon) = \left(1-\Phi\left(\frac{\epsilon-\mu}{\sigma}\right)\right)^N$. To satisfy the conditions of the lemma, we should find the first N for which $\left(1-\Phi\left(\frac{\epsilon-\mu}{\sigma}\right)\right)^N \leq \delta$ holds that means $N \geq \left[\frac{log\ (\delta)}{log\ (1-\Phi(\frac{\epsilon-\mu}{\sigma}))}\right]$.

We can use MAE instead of MRE in this lemma and have the same results. As a result, Lemma 2 can be used as a statistical criterion to estimate the upper bound for the number of random policies to find a near optimal one with the provided level of confidence. Having the values of δ and ϵ as the parameters of CI and considering μ and σ as the mean and standard deviation for a set of random policies, we can determine the required upper bound as $\left[\frac{log\ (\delta)}{log\ (1-\Phi(\frac{\epsilon-\mu}{\sigma}))}\right]$.

Notice that for applying SMC for MDP models and for a given confidence level $1-\delta$ and margin error ϵ as the conditions of CI, we should consider two steps to satisfy these conditions: the first step should select a sample of policies based on the given parameters, and the second step should run sufficient simulations to approximate the required reachability probabilities. However, these steps are independent and each has its own error rate. Hence, we need to consider two error rates ϵ_1 and ϵ_2 such that they satisfy the overall conditions of CI. In this paper, we consider $\epsilon_1 = \epsilon_2 = \frac{\epsilon}{2}$ to determine the values of N and M as the number of policies and simulation runs.

Example 3. We propose the PRISM program for a small MDP with 8 states, as shown in Figure 2. The first 6 states of this MDP have exactly two actions. The last state (where s = 8) is a goal state, and its predecessor is a dead state that cannot reach any other state.

```
mdp
module m
s : [1..8];
[] s = 1 -> (s' = 2);
[] s <= 2 -> .5:(s' = 4) + .5:(s' = 6);
[] s = 2 -> 1/3:(s' = 1) + 1/3:(s' = 2) + 1/3:(s' = 3);
[] s > 2 & s <= 6 -> .1:(s' = s-2) + .2:(s' = s-1) +
          .3:(s' = s+1) + .4:(s' = s+2);
[] s = 3 -> .25:(s' = 1) + .5:(s' = 4) + .25:(s' = 7);
[] s = 4 | s = 5 -> .2:(s' = s - 3) + .2:(s' = s - 1) +
          .2:(s' = s + 1) + .2:(s' = 7) + .2:(s' = 8);
[] s = 6 -> .5: (s' = 3) + .3:(s' = 4) + .2:(s' = 8);
[] s >= 7 -> (s' = s);
endmodule
label "goal" = s = 8;
```

Figure 2. The PRISM program of an MDP model with 8 states

For this small example, we consider all $2^6 = 64$ possible policies and compute MRE and MAE values for each policy, using the optimal values of each state. To do so, for each policy, we consider the induced DTMC, and for each of the 6 states (excluding the goal and dead states), we run 1000 simulations to approximate the probability of reaching the goal state from that state. The results for the first 20 policies are proposed in Figure 3. In this MDP model, for MAE we have $\mu = .204$ and $\sigma = .073$ and for MRE we have $\mu = .29$ and $\sigma = .105$. For these values and considering $\delta = .05$ and $\epsilon = .02$ as the parameter values for CI, we can use Lemma 2 and compute an upper bound for the number of policies as $N > \frac{log(.05)}{log(1-\Phi(\frac{.02-.29}{.105}))} = 597.64$. Hence, to satisfy the CI conditions and given the specified meanand standard deviation of MAE, we need to consider at least 598 policies.

MAE	MRE
0.208134	0.299456
0.226751	0.325582
0.172228	0.24458
0.230932	0.33209
0.239937	0.340113
0.281499	0.402693
0.239313	0.343651
0.090376	0.128257
0.159229	0.226133
0.208134	0.299456
0.116892	0.165172
0.281499	0.402693
0.227049	0.326282
0.186122	0.264224
0.172228	0.24458
0.129896	0.183759
0.281723	0.399066
0.138241	0.195554
0.244652	0.351687
0.116892	0.165172

Figure 3. The computed MAE and MRE for the first 20 policies for MDP model of Example 2

Although Lemma 2 proposes a straightforward method to compute an upper bound for the number of random policies to find a near optimal one for SMC, several challenges exist for applying it for large MDP model:

- It supposes that the distribution of MRE (or MAE) is normal. Otherwise, the proposed upper-bound is not reliable. This condition is verifiable by using some standard tests (Kolmogorov-Smirnov test [33] for example) for probability distributions.

- It is not easy to compute the exact values for μ and σ (called true standard deviation in this section). Using a random sample of policies, one can estimate these quantities with a specified confidence level. These errors in approximating μ and σ should be considered in Lemma 2 to have a more conservative upper bound.

- Computing MAE and MRE relies on having exact values of x_i for each state $s_i \in S$. For feasible models, an exhaustive model checker (e.g., PRISM) can apply the interval iteration method to compute these values with the desired precision. For such models, this lemma serves as a benchmark for assessing the feasibility of using SMC. For large models that expose the state explosion problem, one can rely on MAE by using an approximation of MOV. In the worst case, we consider $x_i = 1$ for all states in the maximal and $x_i = 0$ for all states in the minimal reachability probabilities. Although this approximation may be far from exact values, it provides a sound upperbound on the number of policies. In the next section, we propose a more realistic solution for this challenge.

- In the case of state explosion, it is not possible to compute x_i^π for all states $s_i \in S$. Instead, we consider a sample set $S_{sampl} \subset S$ of m states and for a given policy $\pi \in Pol_M$ our method uses $\frac{1}{m}\sum_{s_i \in S_{samples}} x_i^\pi$ as an approximation for MOV. To approximate each x_i^π it applies SMC on the induced DTMC. In this approach, it is better to consider MAE if $\frac{1}{n}\sum_{i=1}^{n} x_i$ can be approximated in some ways. This solution introduces two new problems in the precision of computations: First, the standard deviation of computed values for the states of $S_{samples}$ may be far from the true standard deviation of MRE (or MAE). Increasing the size of $S_{samples}$ may reduce this error. Unfortunately, there is not any simple method to determine an appropriate size for the sample set $S_{samples}$. Our experiments show that this standard deviation is at most two times larger than the true standard deviation if we use $m = 1000$. Second, using SMC for approximating reachability probabilities of states of $S_{samples}$ introduces some errors in the computed standard deviation. Our solution to approximate the true standard deviation is to increase the number of simulations. Based on the proposed parameters ϵ and δ of lemma 2 and the approximated standard deviation of the state-values of $S_{samples}$, one can determine a bound for errors of SMC on the induced DTMC and propose an upper bound for the number of simulations for each state of $S_{samples}$. Although our solutions to these two challenges burden some overhead, it is possible to apply a multi-processor to accelerate these computations. In addition, some improved methods for SMC of DTMCs are available and can be used to enhance these two solutions [16].

- In normal distribution, a random variable can have any value in $[-\infty, +\infty]$. However, we have $0 \leq MRE \leq 1$.

For the sake of simplicity, one can disregard the cases where in normal distribution$MRE > 1$ or $MRE < 0$. Otherwise, it is possible to consider truncated normal distribution and extend the results of this section to this distribution.

In this section, we propose a method for computing an upper bound on the number of randomly selected policies required to achieve the desired precision in SMC for reachability probabilities in MDPs. We reviewed the challenges of using Lemma 2 and proposed solutions. This method can be applied to any set of policies that induces a normal distribution of MRE and MAE values. Otherwise, given a hypothesis about the probability distribution of these random variables, some modifications are required to apply Lemma 2.

In general, the time complexity of our proposed technique for determining an upper bound on the number of policy samples is independent of the size of the MDP models. It mainly depends on the number of policies for statistical analysis and also on ϵ and δ as the parameters of confidence interval. Considering a constant memory consumption for representing each policy, the space complexity of our approach is in $O(1)$.

4. Experimental Results

To evaluate our proposed approaches, we consider six case-study classes from the PRISM and STORM benchmarks [27, 33]. Brief information on these case studies, including their names, parameter values, numbers of states, actions, and transitions, is presented in Table 1. We use PRISM 4.6, the most recent available version, to implement these approaches. The results of our experiments are proposed in Table 2. In this table, we present the approximate mean and standard deviation of MRE and MAE, as defined in Equations1 and2. Depending on the running times, we consider between 100 and 500 samples for these approximations. We use the standard value-iteration method [1] to compute state reachability probabilities. We apply lemma 2 by setting $\delta = 0.01$ and $\epsilon = 0.01$to compute an upper bound for N as the size of a sample set of policies. The upper bounds can be evaluated in practice by applying Lemma 2 and using the computed parameter values. These upper bounds can be more precise if one can compute exact state values in advance. To do so, we use the interval iteration method in PRISM to compute precise values of states. The results of Table 2 demonstrate that the computed upper bounds are so huge that it is not easy to verify them directly. However, they primarily rely on Lemma 2 and are applicable when state values are largely independent.

Table 1. Information of the selected MDP models

Model	Parameter value	|S|	|Act|	|Trans|
Coin3	K=20	25184	52224	65220
Coin4	K=20	206976	558208	697312
Firewrite	ddl=1500	853408	1074328	1177444
Firewrite (ddl=4000)	ddl=4000	2493408	3146828	3444944
Mer	n=2000	11815564	45370339	46540354
Mer	n=4000	23629564	90734339	93074354
Zeroconf	K=10	3001911	5520579	6787615
Zeroconf	K=18	5477150	10095684	12374708
Wlan3	TTM=2500	1082342	2125690	2206536
Wlan3	TTM=4500	1874342	3733690	3814536
Wlan tb3 (TTM=10)	deadline=100	4767507	6160793	10160675
Wlan tb3 (TTM=10)	deadline=150	8941451	11504363	19164337

Table 2. Computed Upper-bound for the number of policies

Model	MRE			MAE		
(parameter)	μ	σ	N	μ	σ	N
Coin3(K=20)	0.285	.02	1e+31	0.024	1.45e-3	4e+22
Coin4(K=20)	0.20	4.98e-3	3e+38	0.020	5.98e-4	3e+31
Firewrite (ddl=1500)	0.423	5.45e-4	2e+787	0.427	5.57e-4	7e+748
Firewrite (ddl=4000)	0.20	1.7e-4	8e+1117	0.02	1.74e-4	2e+58
Mer(n=2000)	0.806	8.02e-3	3e+99	0.139	1.91e-4	5e+676
Mer(n=4000)	0.806	1.58e-4	6e+5037	0.139	1.58e-4	1e+816
Zeroconf(K=10)	0.938	6.34e-3	8e+146	0.49	1.7e-3	9e+282
Zeroconf(K=18)	0.969	7.18e-3	1e+133	0.573	1.18e-3	4e+477
Wlan3 (TTM=2500)	0.363	9.43e-2	5750	8.7E-3	1.6e-3	9
Wlan3 (TTM=4500)	0.374	0.108	1375	8.84E-3	1.76e-3	8
Wlan tb3 (dl=100)	0.765	3.14e-4	6e+2404	0.165	8.72e-4	6e+177

Wlan tb3 (dl=150)	0.55	3.97e-4	9e+1360	0.137	8.3e-4	8e+153

In most cases, the approximated sigma values are so small that the proposed upper bound on the number of policies exceeds 1e+30. The only exception is the WLAN case studies, where this upper bound is small, showing that LSS can approximate reachability probabilities for these models with the desired level of confidence. Although these results are disappointing in some cases and the provided upperbounds are so large that they are not useful in practice, they show that more advanced techniques (such as machine learning methods) are needed to guide SMC toward faster convergence to optimal values.

We use the smart sampling method supported by the Plasmalab model checker as the baseline and compare the precision of the computed values with that of our method. To do so, we consider our method while setting$\delta = 0.01$ and $\epsilon = 0.01$ as the parameters for CI and the default parameters for Plasmalab model checker. We report absolute errors for the results of running smart sampling over case studies in Table 3. While we expect small errors with high confidence, the reported errors are high in several cases where the smart sampling method is applied.

Table 3. **Absolute errors of running smart sampling as baseline**

Model	Parameter value	Absolute error
Coin3	K=20	0.134
Coin4	K=20	0.087
Firewrite	ddl=1500	0.453
Firewrite (ddl=4000)	ddl=4000	0.521
Mer	n=2000	0.032
Mer	n=4000	0.046
Zeroconf	K=10	0
Zeroconf	K=18	0
Wlan3	TTM=2500	0.012
Wlan3	TTM=4500	0.024
Wlan tb3	delay=100	0.174
Wlan tb3	delay=150	0.218

5. Conclusion and Future Works

In this paper, we propose an approach to determine an upper bound on the number of policies required to guarantee that the conditions of a given CI are satisfied when applying SMC to verify the unbounded reachability probabilities of MDPs. Some challenges remain for future work. Specifically, it is important to investigate how the structure of an MDP model can affect the applicability of the central limit theorem to our proposed techniques. Another important research direction is to develop efficient machine learning techniques with low space complexity to predict near-optimal policies for large models. These techniques can use neural networks or random forests to prioritize actions, thereby accelerating convergence to optimal policies. Using such techniques in our approach can yield a higher mean value and reduce the number of policies required to satisfy the CI conditions.

6. References

[1] Baier, C., &Katoen, J. P. (2008). Principles of model checking. MIT press, Cambridge, United States.

[2] Baier, C., Hermanns, H., &Katoen, JP. (2019). The 10,000 Facets of MDP Model Checking. Computing and Software Science. Lecture Notes in Computer Science, vol 10000. Springer, Cham, Switzerland. doi:10.1007/978-3-319-91908-9_21.

[3] Hartmanns, A., Junges, S., Quatmann, T., &Weininger, M. (2023). A Practitioner's Guide to MDP Model Checking Algorithms. Tools and Algorithms for the Construction and Analysis of Systems. TACAS 2023. Lecture Notes in Computer Science, vol 13993. Springer, Cham, Switzerland.doi:10.1007/978-3-031-30823-9_24.

[4] Hartmanns, A., &Kaminski, B.L. (2020). Optimistic Value Iteration. Computer Aided Verification. CAV 2020. Lecture Notes in Computer Science, vol 12225. Springer, Cham, Switzerland. doi:10.1007/978-3-030-53291-8_26.

[5] Younes, H.L.S., &Simmons, R.G. (2002). Probabilistic Verification of Discrete Event Systems Using Acceptance Sampling. Computer Aided Verification. CAV 2002. Lecture Notes in Computer Science, vol 2404. Springer, Berlin, Germany. doi:10.1007/3-540-45657-0_17.

[6] D'Argenio, P., Legay, A., Sedwards, S., &Traonouez, L. M. (2015). Smart sampling for lightweight verification of markov decision processes. International Journal on Software Tools for Technology Transfer, 17(4), 469–484. doi:10.1007/s10009-015-0383-0.

[7] Hartmanns, A.: On the analysis of stochastic timed systems. Ph.D. thesis, Saarland University, Germany (2015)

[8] Legay, A., Sedwards, S., &Traonouez, LM. (2015). Scalable Verification of Markov Decision Processes. In: Canal, C., Idani, A. (eds) Software Engineering and Formal Methods. SEFM 2014. Lecture Notes in Computer Science, vol 8938.

Springer, Cham, Switzerland. doi:10.1007/978-3-319-15201-1_23.

[9] Kanav, S., Křetínský, J., Larsen, K.G. (2025). Statistical Model Checking the 2024 Edition!.Bridging the Gap Between AI and Reality. AISoLA 2024. Lecture Notes in Computer Science, vol 15217. Springer, Cham, Switzerland. doi:10.1007/978-3-031-75434-0_21.

[10] Bogdoll, J., Hartmanns, A., &Hermanns, H. (2012). Simulation and Statistical Model Checking for Modestly Nondeterministic Models. Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance. MMB&DFT 2012. Lecture Notes in Computer Science, vol 7201. Springer, Berlin, Germany. doi:10.1007/978-3-642-28540-0_20.

[11] Henriques, D., Martins, J. G., Zuliani, P., Platzer, A., & Clarke, E. M. (2012). Statistical Model Checking for Markov Decision Processes. 2012 Ninth International Conference on Quantitative Evaluation of Systems, 84–93. doi:10.1109/qest.2012.19.

[12] Budde, C. E., D'Argenio, P. R., Hartmanns, A., &Sedwards, S. (2020). An efficient statistical model checker for nondeterminism and rare events. International Journal on Software Tools for Technology Transfer, 22(6), 759–780. doi:10.1007/s10009-020-00563-2.

[13] Budde, C.E., Hartmanns, A., Meggendorfer, T., Weininger, &M., Wienhöft, P. (2025). Sound Statistical Model Checking for Probabilities and Expected Rewards. Tools and Algorithms for the Construction and Analysis of Systems. TACAS 2025. Lecture Notes in Computer Science, vol 15696. Springer, Cham, Switzerland. doi:10.1007/978-3-031-90643-5_9.

[14] Parmentier, M., Legay, A., Chenoy, F. (2024). Optimized Smart Sampling. Bridging the Gap Between AI and Reality. AISoLA 2023. Lecture Notes in Computer Science, vol 14380. Springer, Cham, Switzerland. doi:10.1007/978-3-031-46002-9_10.

[15] Kwiatkowska, M., Norman, G., &Segala, R. (2001). Automated Verification of a Randomized Distributed Consensus Protocol Using Cadence SMV and PRISM. Computer Aided Verification. CAV 2001. Lecture Notes in Computer Science, vol 2102. Springer, Berlin, Germany. doi:10.1007/3-540-44585-4_17.

[16] Reijsbergen, D., de Boer, P.-T., Scheinhardt, W., &Haverkort, B. (2014). On hypothesis testing for statistical model checking. International Journal on Software Tools for Technology Transfer, 17(4), 377–395. doi:10.1007/s10009-014-0350-1.

[17] Okamoto, M. (1959). Some inequalities relating to the partial sum of binomial probabilities. Annals of the Institute of Statistical Mathematics, 10(1), 29–35. doi:10.1007/BF02883985.

[18] Kearns, M., Mansour, Y., & Ng, A. Y. (2002). A sparse sampling algorithm for near-optimal planning in large Markov decision processes. Machine Learning, 49(2–3), 193–208. doi:10.1023/A:1017932429737.

[19] Ashok, P., Křetínský, &J., Weininger, M. (2019). PAC Statistical Model Checking for Markov Decision Processes and Stochastic Games. Computer Aided Verification. CAV 2019. Lecture Notes in Computer Science, vol 11561. Springer, Cham, Switzerland. doi:10.1007/978-3-030-25540-4_29.

[20] Budde, C.E., D'Argenio, P.R., Fraire, J.A., Hartmanns, A., Zhang, Z. (2025). Modest Models and Tools for Real Stochastic Timed Systems. Principles of Verification: Cycling the Probabilistic Landscape. Lecture Notes in Computer Science, vol 15261. Springer, Cham, Switzerland. doi:10.1007/978-3-031-75775-4_6.

[21] Rataj, A., &Woźna-Szcześniak, B. (2018). Extrapolation of an Optimal Policy using Statistical Probabilistic Model Checking.FundamentaInformaticae, 157(4), 443–461. doi:10.3233/FI-2018-1637.

[22] Gros, T.P., Hermanns, H., Hoffmann, J., Klauck, M., &Steinmetz, M. (2020). Deep Statistical Model Checking. Formal Techniques for Distributed Objects, Components, and Systems. FORTE 2020. Lecture Notes in Computer Science, vol 12136. Springer, Cham, Switzerland. doi:10.1007/978-3-030-50086-3_6.

[23] Kwiatkowska, M., Norman, G., &Parker, D. (2011). PRISM 4.0: Verification of Probabilistic Real-Time Systems. Computer Aided Verification. CAV 2011. Lecture Notes in Computer Science, vol 6806. Springer, Berlin, Germany. doi:10.1007/978-3-642-22110-1_47.

[24] Agarwal, C., Guha, S., Křetínský, J., &Pazhamalai, M. (2025). PAC statistical model checking of mean payoff in discrete- and continuous-time MDP. Formal Methods in System Design, 66(2), 195–237. doi:10.1007/s10703-024-00463-0.

[25] Perez, M., Somenzi, F., & Trivedi, A. (2024). A PAC Learning Algorithm for LTL and Omega-Regular Objectives in MDPs. Proceedings of the AAAI Conference on Artificial Intelligence, 38(19), 21510–21517. doi:10.1609/aaai.v38i19.30148.

[26] Azeem, M., Chakraborty, D., Kanav, S., Křetínský, J., Mohagheghi, M., Mohr, S., &&Weininger, M. (2025). 1–2–3–Go! Policy Synthesis for Parameterized Markov Decision Processes via Decision-Tree Learning and Generalization. Verification, Model Checking, and Abstract Interpretation. VMCAI 2025. Lecture Notes in Computer Science, vol 15530. Springer, Cham, Switzerland. doi:10.1007/978-3-031-82703-7_5

[27] Clarke, E. M., Henzinger, T. A., Veith, H., &Bloem, R. (Eds.). (2018). Handbook of Model Checking. Springer International Publishing, Cham, Switzerland. doi:10.1007/978-3-319-10575-8.

[28] D'argenio, P. R., Fraire, J., Hartmanns, A., &Raverta, F. (2025). Comparing Statistical, Analytical, and Learning-Based Routing Approaches for Delay-Tolerant Networks. ACM Transactions on Modeling and Computer Simulation, 35(2). doi:10.1145/3665927.

[29] Meggendorfer, T., Weininger, M., &Wienhöft, P. (2026). What Are the Odds? Improving Statistical Model Checking

of Markov Decision Processes. Quantitative Evaluation of Systems and Formal Modeling and Analysis of Timed Systems. QEST+FORMATS 2025. Lecture Notes in Computer Science, vol 16143. Springer, Cham, Switzerland. doi:10.1007/978-3-032-05792-1_11

[30] Legay, A., Sedwards, S., &Traonouez, LM. (2016). Plasma Lab: A Modular Statistical Model Checking Platform. Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques. ISoLA 2016. Lecture Notes in Computer Science(), vol 9952. Springer, Cham, Switzerland. doi:10.1007/978-3-319-47166-2_6.

[31] Safari, M. (2025). 'Experimental analysis, statistical modeling and optimization of the edge effects associated with laser-bent perforated sheets', Contributions of Science and Technology for Engineering, 2(3), pp. 37-44. doi: 10.22080/cste.2025.28804.1019

[32] Hensel, C., Junges, S., Katoen, J. P., Quatmann, T., & Volk, M. (2022). The probabilistic model checker Storm. International Journal on Software Tools for Technology Transfer, 24(4), 589–610. doi:10.1007/s10009-021-00633-z.

[33] Stephens, M. A. (1974). EDF statistics for goodness of fit and some comparisons. Journal of the American Statistical Association, 69(347), 730–737. doi:10.1080/01621459.1974.10480196.