Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

COSIMLA WITH GENERAL REGENERATION SET TO COMPUTE MARKOV CHAIN STATIONARY EXPECTATIONS

Peter W. Glynn

Department of Management Science and Engineering
Stanford University
475 Via Ortega
Stanford, CA 94305, USA

Zeyu Zheng

Department of Industrial Engineering and
Operations Research
University of California, Berkeley
4125 Ethcheverry HAll
Berkeley, CA 94709, USA

ABSTRACT

We extend the COSIMLA approach (short for "COmbined SIMulation and Linear Algebra") recently developed in Zheng, Infanger, and Glynn (2022) to compute stationary expectations for Markov chains with large or infinite discrete state space. Our work follows the idea of combing the best of linear algebra and simulation—using linear algebra to compute the "center" of the state space and using simulation to compute the contributions from outside of the "center". Different from Zheng, Infanger, and Glynn (2022) that needed to fix a single regeneration state, our work develops a new method that allows the use of a flexible regeneration set with a finite number of states. We show that this new method allows more efficient computation for the COSIMLA approach.

1 INTRODUCTION

Many modeling applications involve discrete state space Markov chains. This paper concerns the computation of steady-state expectations, or equivalently saying, stationary expectations, for Markov chains with a discrete state space. When the state space is finite and has a moderate size, one can compute the stationary distribution through numerically solving a system of linear equations, and then compute the stationary expectations through a linear algebra operation. This is well documented, for example, in Heyman and Sobel (2004) and Asmussen (2008). However, when the state space becomes large or includes countably infinite many states, such computation becomes intractable or impossible.

On the other hand, Monte Carlo simulation can be used to compute stationary expectations for discrete space Markov chains. The use of simulation is feasible even when the state space is infinite. The challenge of simulation is the excessive amount of replications needed, especially when one desires a high accuracy.

A recent work Zheng, Infanger, and Glynn (2022) proposes the method of COSIMLA, short for "COmbined SIMulation and Linear Algebra". That paper provides the first efficient numerical method that combines the best aspects of numerical linear algebra and simulation. For COSIMLA, numerical linear algebra is used to analyze the "center" of the state space, while simulation is used to estimate contributions to the stationary expectations from path excursions outside the "center". In particular, COSIMLA can be viewed as using simulation to enhance the truncation-based linear algebra method for Markov chain computations; see Seneta (1980), Kuntz, Thomas, Stan, and Barahona (2021) for prior work on truncation-based approximation schemes for Markov chains with infinite or large state space.

In this work, we extend the COSIMLA approach discussed in Zheng, Infanger, and Glynn (2022) for the computation of Markov chain stationary expectations. That paper proposes a COSIMLA approach that uses path excursions that start from a fixed regeneration state z, and stop when the chain returns to z. In this work, we extend the COSIMLA approach, so that the method can exploit path excursions that

979-8-3503-6966-3/23/$31.00 ©2023 IEEE

start in a finite subset K contained within A, and that stop when the chain returns to K. This extension can sometimes significantly reduce the simulation's computational effort, since the simulation is no longer required to follow the path all the way back to the state z, and can instead terminate when it hits K.

2 COMPUTATIONAL TASK AND SETTING

We are concerned with the computation of stationary expectations for Markov chains. The setting is as follows. Denote $X = (X_n : n \geq 0)$ as an irreducible positive recurrent Markov chain with discrete state space S. Denote $P = (P(x,y) : x,y \in S)$ as the one step transition matrix. Due to irreducible positive recurrent, there exists a unique stationary distribution, denoted as $\pi = (\pi(x) : x \in S)$, encoded as a row vector. The stationary distribution π satisfies the following linear system of equations

$$\pi = \pi P.$$

Denote $r = (r(x) : x \in S)$ as a reward function, encoded as a column vector. Without loss of generality, we presume that $r(x) \in \mathbb{R}_+$ for each $x \in S$. In this work, we consider a computation task where P and r are given and the goal is to compute the stationary expectation

$$\pi r = \sum_{x \in S} \pi(x) r(x).$$

When the state space S is small, this computation would be a simple two-step procedure. First, one solves for π from the linear system of equations $\pi = \pi P$. Then, one computes the inner product of π and r through a simple linear algebra operation. However, when the state space S becomes large or infinite, such procedure will become computationally intractable or impossible. This work concerns the computation task when the state space S is large or infinite.

3 OUR APPROACH

Choose $K \subseteq A \subseteq S$ with $|A| < \infty$, where K is a non-empty "regeneration" subset, and A, the "truncation" subset, has a size such that linear systems involving $|A|$ equations and $|A|$ unknowns can be tractably numerically computed. Denote $A' = A - K = \{x \in S : x \in A, x \notin K\}$. The follow plot illustrates the relative connections among the set K, set A, and set S. Figure 1 gives an illustration of the relationships of these sets.

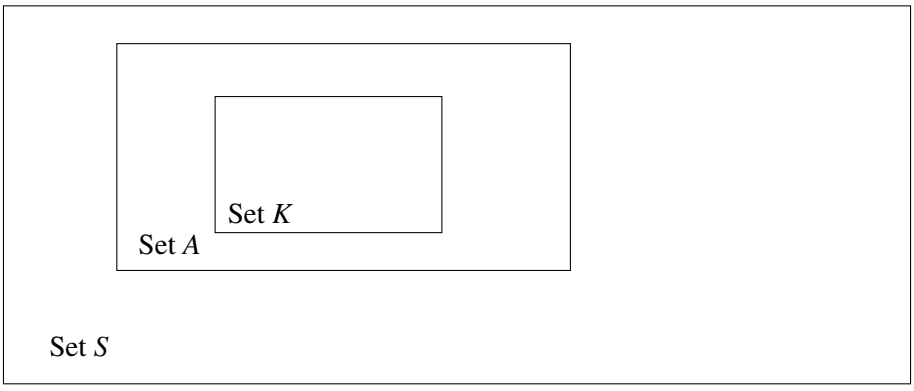

Figure 1: Illustration of sets S, A, and K.

For any $x \in S$, denote $P_x(\cdot)$ be the probability on the space of X conditioned on $X_0 = x$. Denote $\mathbb{E}_x(\cdot)$ as its associated expectation. We specify three items of notation to facilitate the discussion of our approach.

- $T_K = \inf\{n > 0 : X_n \in K\}$
- $T = \inf\{n \geq 0 : X_n \in A^c\}$
- $D = \{y \in A^c : P_x(X_T = y) > 0 \text{ for some } x \in A\}$

For $x, y \in K$, define $P_K = (P_K(x,y) : x,y \in K)$ as $P_K(x,y) = P_x(X_{T_K} = y)$. Let $\pi_K = (\pi_K(x) : x \in K)$ be a distribution on K such that $\pi_K P_K = \pi_K$. With the goal to compute $\alpha = \sum_{x \in S} \pi(x) r(x)$, we note that

$$\alpha = \frac{\mathbb{E}_{\pi_K} \sum_{j=0}^{T_K - 1} r(X_j)}{\mathbb{E}_{\pi_K} T_K},$$

where $\mathbb{E}_{\pi_K}(\cdot)$ is denoted such that for any non-negative function g defined on the path-space of X, $\mathbb{E}_{\pi_K}(g(X)) = \sum_{x \in K} \pi_K(x) \mathbb{E}_x(g(X))$.

For $x \in K$, put

$$\begin{aligned} u(x) &= \mathbb{E}_x \sum_{j=0}^{T_K - 1} r(X_j) \\ &= \mathbb{E}_x \sum_{j=0}^{(T \wedge T_K) - 1} r(X_j) + \sum_{y \in D} P_x(X_T = y, T < T_K) \, \mathbb{E}_y \sum_{j=0}^{T_K - 1} r(X_j) \\ &\triangleq a(r,x) + \sum_{y \in D} F(x,y) \tilde{w}(r,y). \end{aligned}$$

That is, we define

$$\begin{aligned} a(r,x) &= \mathbb{E}_x \sum_{j=0}^{(T \wedge T_K) - 1} r(X_j), \\ F(x,y) &= P_x(X_T = y, T < T_K), \\ \tilde{w}(r,y) &= \mathbb{E}_y \sum_{j=0}^{T_K - 1} r(X_j). \end{aligned}$$

We also write that $a(r) = (a(r,x) : x \in K)$, encoded as a column vector, $F = (F(x,y) : x \in K, y \in D)$ as a $|K| \times |D|$ matrix, and $\tilde{w}(r) = (\tilde{w}(r,y), y \in D)$ as a column vector.

In order to compute the numerator and denominator, we also need to compute π_K. Note that

$$\begin{aligned} P_K(x,y) &= P_x(X_{T_K} = y) \\ &= P_x(X_{T_K} = y, T_K < T) + \sum_{z \in D} P_x(X_T = z, T < T_K) P_z(X_{T_K} = y) \\ &\triangleq \tilde{A}(x,y) + \sum_{z \in D} F(x,z) \tilde{H}(z,y). \end{aligned}$$

That is, we denote

$$\begin{aligned} \tilde{A}(x,y) &= P_x(X_{T_K} = y, T_K < T), \\ \tilde{H}(z,y) &= P_z(X_{T_K} = y), \end{aligned}$$

and write $\tilde{A} = (\tilde{A}(x,y) : x,y \in K)$ and $\tilde{H} = (\tilde{H}(z,y) : z \in D, y \in K)$ as the corresponding matrices.

Next we compute $a(r)$, F, and $\tilde{A}$ through linear algebra operations. For $a(r) = (a(r,x) : x \in K)$, we observe that for any $x \in K$,

$$\begin{aligned} a(r,x) &= \mathbb{E}_x \sum_{j=0}^{(T \wedge T_K)-1} r(X_j) \\ &= r(x) + \sum_{y \in A'} P(x,y)\, \mathbb{E}_y \sum_{j=0}^{(T \wedge T_K)-1} r(X_j). \end{aligned}$$

If we denote

$$\begin{aligned} r_1 &= (r(x) : x \in K) \\ r_2 &= (r(x) : x \in A') \\ P_{11} &= (P(x,y) : x,y \in K) \\ P_{12} &= (P(x,y) : x \in K, y \in A') \\ P_{22} &= (P(x,y) : x,y \in A'), \end{aligned}$$

we have that

$$a(r) = r_1 + P_{12}(I - P_{22})^{-1} r_2,$$

Also,

$$F = P_{13} + P_{12}(I - P_{22})^{-1} P_{23},$$

where

$$\begin{aligned} P_{13} &= (P(x,y) : x \in K, y \in D) \\ P_{23} &= (P(x,y) : x \in A', y \in D). \end{aligned}$$

Also,

$$\tilde{A} = P_{11} + P_{12}(I - P_{22})^{-1} P_{21},$$

where

$$P_{21} = (P(x,y) : x \in A', y \in K).$$

Fix $e = (e(x) : x \in S)$ where $e(x) \equiv 1$ for all x. For the terms $\tilde{w}(r)$, $\tilde{w}(e)$, and $\tilde{H}$, we compute by Monte Carlo simulation. That is, for each $y \in D$, we simulate n independent sample paths of X conditional on $X_0 \in D$, where each path stops at T_K, the regeneration time to set K. Denote $\tilde{w}_n(r)$, $\tilde{w}_n(e)$, $\tilde{H}_n$, respectively, for the estimators for $\tilde{w}(r)$, $\tilde{w}(e)$, $\tilde{H}$ using n independent replications of simulation.

Now we can have an estimator for P_K, given by

$$\hat{P}_K = \tilde{A} + F\tilde{H}_n.$$

We compute a distribution π_n on K as

$$\pi_n = \pi_n \hat{P}_K.$$

So, we arrive at an estimator for α, given by

$$\alpha_n = \frac{\pi_n(a(r) + F\tilde{w}_n(r))}{\pi_n(a(e) + F\tilde{w}_n(e))}. \tag{1}$$

For $x, y \in K$, define $\Pi_K = (\Pi_K(x,y) : x,y \in K)$ as $\Pi_K(x,y) = \pi_K(y)$. That is, Π_K is a matrix with size $|K| \times |K|$ where each row is identical to π_K. We note that

$$\pi_n - \pi = \pi(\hat{P}_K - P_K)(I - P_K + \Pi_k)^{-1} = \pi F(\tilde{H}_n - H)(I - P_K + \Pi_k)^{-1}.$$

So,

$$\alpha_n - \alpha = \frac{\pi F(\tilde{w}_n(r) - \alpha\tilde{w}_n(e)) + \pi F(\tilde{H}_n - H)(I - P_K + \Pi_k)^{-1}(a(r) - \alpha a(e) + F(\tilde{w}(r) - \alpha\tilde{w}(e)))}{\pi(a(e) + F\tilde{w}(e))} + o_p(n^{-1/2}),$$

which enables us to construct central limit theorem for the estimator.

4 NUMERICAL EXAMPLE

In this section, we present two parts of numerical experiments to demonstrate the performance of our approach for computing Markov chain stationary expectations. In the first part, we consider a one-dimensional birth-death Markov chain. In the second part, we consider a two-dimensional queueing network. In particular, we demonstrate the comparison of our approach that allows a flexible size of regeneration set, compared to the approach in Zheng, Infanger, and Glynn (2022) that only allows the regeneration set to contain a single state. Zheng, Infanger, and Glynn (2022) conducted numerical experiments on a one-dimensional birth-death Markov chain, but not on a two-dimensional queueing network.

4.1 One-dimensional Brith-death Markov Chain

For this subsection, we adopt the same underlying model and notation as Zheng, Infanger, and Glynn (2022). For completeness, we re-iterate the model setup. Consider a discrete-time Markov chain on $S = \mathbb{Z}_+$, in which the state $X = (X_n : n \geq 0)$ evolves according to the stochastic recursion

$$X_{n+1} = [X_n + Z_{n+1} - 1]^+,$$

where $[x]^+ \triangleq \max(x, 0)$ and the Z_n's are independent and identically distributed (iid) with $P(Z_1 = 0) = q$, $P(Z_1 = 1) = 1 - p - q$, and $P(Z_1 = 2) = p$. This Markov chain can describe the number-in-system process for a slotted time queue in which the system can serve 1 customer in each time slot, and Z_{n+1} represents the number of customers arriving at the beginning of slot $n+1$. The one step transition matrix P of X is a tri-diagonal matrix given by

$$P = \begin{pmatrix} 1-p & p & 0 & 0 & 0 & \cdots \\ q & 1-(p+q) & p & 0 & 0 & \ldots \\ 0 & q & 1-(p+q) & p & 0 & \ldots \\ 0 & 0 & q & 1-(p+q) & p & \\ \vdots & \vdots & & \ddots & \ddots & \ddots \end{pmatrix},$$

where $q > p > 0$ and $p + q \leq 1$.

For parameter settings, we adjust the ratio $\rho \triangleq p/q$ where $p + q = 1$. We adjust two parameters k and M, and consider the truncation set $A = \{0, 1, \ldots, M\}$ and the regeneration set as $K = \{0, 1, \ldots, k\}$. We use n to demonstrate the number of independent replications for the simulation part of the estimator.

For each given parameter setting, we present the mean square error (MSE) and the mean wall-clock time to generate one estimator. The wall-clock time, which is sometimes referred to as elapsed real time, is the actual time taken by the computer to compute an estimator. The means are computed via 1000 replications for each estimator.

4.1.1 Small Truncation Set

We first present a group of numerical results when the truncation set M is relatively small, taking value in the range of 10 to 500. In this case, the time for linear algebra computation is much smaller compared to the time for simulation. We focus on the comparison across different choices of k for the regeneration set as $K = \{0, 1, \ldots, k\}$. A larger k then means a larger regeneration set.

Table 1 to Table 6 present results from a mixed choices of ρ, M, n, and each table includes multiple choices of k. Particularly, note that when $k = 0$, it exactly corresponds to the estimator given by Zheng, Infanger, and Glynn (2022). We have the following core observation.

- When M is relatively small, for fixed ρ, M, n, the larger k is, the less mean wall-clock time the algorithm takes, while maintaining a stable level of MSE.

Table 1: Estimator performance with $\rho = 0.8$, $M = 10$, $n = 1000$.

Estimator/Performance	Mean squared error	Mean wall-clock time (seconds)
$k = 0$	4.81×10^{-3}	0.136
$k = 1$	5.19×10^{-3}	0.099
$k = 2$	4.53×10^{-3}	0.095
$k = 3$	5.18×10^{-3}	0.086
$k = 4$	5.24×10^{-3}	0.083

Table 2: Estimator performance with $\rho = 0.95$, $M = 100$, $n = 200$.

Estimator/Performance	Mean squared error	Mean wall-clock time (seconds)
$k = 0$	9.45×10^{-3}	0.717
$k = 5$	9.37×10^{-3}	0.698
$k = 10$	5.98×10^{-3}	0.672
$k = 15$	8.14×10^{-3}	0.649
$k = 20$	1.01×10^{-2}	0.623

Table 3: Estimator performance with $\rho = 0.8$, $M = 20$, $n = 250$.

Estimator/Performance	Mean squared error	Mean wall-clock time (seconds)
$k = 0$	7.69×10^{-4}	0.049
$k = 2$	8.54×10^{-4}	0.047
$k = 4$	7.81×10^{-4}	0.041
$k = 6$	7.64×10^{-4}	0.036
$k = 8$	7.54×10^{-4}	0.032
$k = 10$	6.90×10^{-4}	0.028
$k = 12$	7.35×10^{-4}	0.023
$k = 14$	7.43×10^{-4}	0.018
$k = 16$	7.77×10^{-4}	0.014

4.1.2 Large Truncation Set

We then consider experiments with relatively large M. Different from the small M situation, now the time for the linear algebra computation starts to match the time for the simulation computation. Because the mean wall-clock time includes both the linear algebra (LA) time and the simulation time, we present related

Table 4: Estimator performance with $\rho = 0.75$, $M = 100$, $n = 200$.

Estimator/Performance	MSE	Mean wall-clock time (seconds)
$k = 0$	1.91×10^{-23}	0.353
$k = 10$	1.85×10^{-23}	0.322
$k = 30$	1.83×10^{-23}	0.243
$k = 50$	1.58×10^{-23}	0.180

Table 5: Estimator performance with $\rho = 0.95$, $M = 200$, $n = 50$.

Estimator/Performance	MSE	Mean wall-clock time (seconds)
$k = 0$	7.78×10^{-6}	0.363
$k = 10$	7.70×10^{-6}	0.351
$k = 20$	7.88×10^{-6}	0.335
$k = 30$	7.89×10^{-6}	0.345
$k = 40$	7.29×10^{-6}	0.299
$k = 50$	7.65×10^{-6}	0.273
$k = 60$	7.83×10^{-6}	0.257
$k = 70$	7.90×10^{-6}	0.240

Table 6: Estimator performance with $\rho = 0.99$, $M = 500$, $n = 50$.

Estimator/Performance	MSE	Mean wall-clock time (seconds)
$k = 0$	1.33	4.77
$k = 25$	1.28	4.47
$k = 50$	1.27	4.13
$k = 75$	1.35	3.99
$k = 100$	1.53	3.83

information. Table 7 presents an illustration of results by varying choices of k. The mean time is computed via 50 replications. As we see from Table 7, as one would anticipate, the linear algebra time increases as k becomes larger and the simulation time decreases as k becomes larger. The mean wall-clock time, in this example, decreases when k becomes larger but seems to stabilize when k becomes closer to M.

Table 7: Estimator performance with $\rho = 0.99$, $M = 2000$, $n = 200$.

Estimator/Performance	MSE	Mean LA time	Mean simulation time	Mean wall-clock time
$k = 500$	1.01×10^{-12}	0.29	73.41	73.70
$k = 1000$	6.81×10^{-13}	1.29	45.89	47.18
$k = 1500$	8.57×10^{-13}	2.98	18.13	21.11
$k = 1700$	3.83×10^{-13}	4.35	13.56	17.90
$k = 1800$	3.22×10^{-13}	4.99	9.65	14.64
$k = 1900$	2.99×10^{-13}	6.09	5.41	11.50
$k = 1925$	3.86×10^{-13}	6.26	4.08	10.33
$k = 1950$	5.33×10^{-13}	7.93	3.64	11.57

4.2 Two-dimensional Queueing Network

In this section, we consider a discrete-time two-node independent queueing network where the state space $S = \mathbb{Z}_+ \times \mathbb{Z}_+$. For any $(a,b) \in S$, the state represents that there are a jobs in node 1 and b jobs in node 2. The jobs transition matrix given by

$$\begin{pmatrix} 0 & 1 & 0 \\ 1-p_1 & 0 & p_1 \\ 1 & 0 & 0 \end{pmatrix}.$$

To be specific, when a job arrives at the network, it will directly go to node 1. When a job departs from node 1, it will go to node 2 with probability p_1 and leave the network with probability $1-p_1$. When a job departs from node 2, it will directly leave the network. Besides, we presume that, in each discrete time period, a job arrives at the network with probability p. The two nodes are each single server queue themselves. The transition probability is given by $P((a,b),(a+1,\max(b-1,0))) = p$, $P((a,b),(\max(a-1,0),\max(b-1,0))) = (1-p_1)(1-p)$ and $P((a,b),(\max(a-1,0),b+1)) = p_1(1-p)$ for arbitrary $(a,b) \in S$. The utilization of the two nodes are given by, respectively, $u_1 = \rho$ and $u_2 = \frac{\rho\rho_1}{1+\rho+\rho_1}$.

We choose the reward function $r = r(x,y) = x+y$, which represents the total jobs in the network. In this case, the stationary expected reward is given by

$$\alpha = \pi r = \frac{u_1}{1-u_1} + \frac{u_2}{1-u_2}. \tag{2}$$

For our approach, we consider two parameters k and M with $0 \le k \le M$. The truncation set is $A = \{(a,b) : a+b \le M\}$ and the regeneration set is $K = \{(a,b) : a+b \le k\}$. When $k = 0$, our approach reduces to that of Zheng, Infanger, and Glynn (2022) with singleton regeneration set. The goal of following numerical results is to compare estimator performance for different choices of k.

Table 8 to Table 12 give the numerical results. We have the following observations.

1. The linear algebra time increases as the choice of k and M increases. Note that when $k = 100$, there are about 5000 different states in the regeneration set. A $k \times k$ matrix then has 24502500 elements. We observe that the linear algebra time can become intractable even when k becomes moderately large.
2. The mean squared error (MSE) does not show an obvious pattern of dependence on the choice of k.

3. As one would anticipate, with all else equal, the larger the k, the longer the mean linear algeba (LA) time and the smaller the mean simulation time.
4. For relatively small utilization rates, we find that the larger k is, the smaller the total wall-clock time.
5. For relatively large utilization rates, we find that when k increases, the mean total wall-clock time may first decrease and then then increase, suggesting that in terms of mean total time, one may find an optimal choice of k that is smaller than M. It seems that such optimal choice of k is likely to be when the mean simulation time equals the mean LA time, but we need more evidence and theory to be able to formally make this claim.

Table 8: Estimator Performance with $M = 10$, $n = 1000$, $u_1 = 0.8$, $u_2 = \frac{2}{7}$.

Estimator/Performance	MSE	Mean wall-clock time	Mean simulation time	Mean LA time
$k = 0$	4.61×10^{-3}	0.661	0.661	8.80×10^{-5}
$k = 1$	4.96×10^{-3}	0.613	0.612	4.57×10^{-4}
$k = 2$	5.32×10^{-3}	0.579	0.578	5.95×10^{-4}
$k = 3$	4.99×10^{-3}	0.511	0.511	3.38×10^{-4}
$k = 4$	5.35×10^{-3}	0.464	0.463	4.10×10^{-4}
$k = 5$	5.13×10^{-3}	0.409	0.408	4.54×10^{-4}
$k = 6$	5.51×10^{-3}	0.359	0.358	5.33×10^{-4}

Table 9: Estimator Performance with $M = 20$, $n = 250$, $u_1 = 0.8$, $u_2 = 0.5$.

Estimator/Performance	MSE	Mean wall-clock time	Mean simulation time	Mean LA time
$k = 0$	8.54×10^{-4}	0.424	0.424	1.45×10^{-4}
$k = 2$	8.34×10^{-4}	0.393	0.392	4.17×10^{-4}
$k = 4$	7.96×10^{-4}	0.369	0.369	3.52×10^{-4}
$k = 6$	7.96×10^{-4}	0.346	0.345	6.99×10^{-4}
$k = 8$	7.97×10^{-4}	0.319	0.316	2.96×10^{-3}
$k = 10$	7.47×10^{-4}	0.293	0.288	4.34×10^{-3}

Table 10: Estimator Performance with $M = 40$, $n = 200$, $u_1 = 0.8$, $u_2 = 0.5$.

Estimator/Performance	MSE	Mean wall-clock time	Mean simulation time	Mean LA time
$k = 0$	6.38×10^{-7}	0.729	0.727	1.91×10^{-3}
$k = 8$	5.51×10^{-7}	0.677	0.672	5.39×10^{-3}
$k = 16$	4.70×10^{-7}	0.594	0.566	0.028
$k = 24$	5.83×10^{-7}	0.522	0.444	0.078
$k = 28$	5.88×10^{-7}	0.510	0.369	0.142
$k = 32$	5.26×10^{-7}	0.520	0.271	0.249
$k = 36$	3.78×10^{-7}	0.597	0.185	0.412
$k = 40$	4.84×10^{-7}	0.735	0.065	0.670

5 CONCLUSION

In this work, we develop a method that extends the COSIMLA approach to allow the use of a general regeneration set. We provide analysis for this method and run numerical experiments to demonstrate the improved efficiency brought by the developed method.

Table 11: Estimator Performance with $M = 50$, $n = 200$, $u_1 = 0.95$, $u_2 = 0.5$.

Estimator/Performance	MSE	Mean wall-clock time	Mean simulation time	Mean LA time
$k = 0$	0.342	6.73	6.72	5.80×10^{-3}
$k = 5$	0.390	6.53	6.52	1.26×10^{-2}
$k = 10$	0.415	5.94	5.92	1.45×10^{-2}
$k = 15$	0.382	5.57	5.55	2.32×10^{-2}
$k = 20$	0.395	4.98	4.93	4.98×10^{-2}
$k = 25$	0.473	4.46	4.36	0.100
$k = 30$	0.439	3.88	3.69	0.189

Table 12: Estimator Performance with $M = 100$, $n = 200$, $u_1 = 0.95$, $u_2 = 0.9$.

Estimator/Performance	MSE	Mean wall-clock time	Mean simulation time	Mean LA time
$k = 0$	0.0093	32.30	32.13	0.167
$k = 10$	0.013	29.34	29.19	0.156
$k = 20$	0.018	28.54	28.36	0.182
$k = 30$	0.015	27.20	26.89	0.308
$k = 40$	0.013	27.60	26.81	0.790
$k = 50$	0.010	27.31	25.26	2.051
$k = 60$	0.0083	28.36	23.40	4.961
$k = 70$	0.012	31.72	20.69	11.03
$k = 80$	0.012	39.20	15.54	23.66
$k = 90$	0.014	53.85	11.01	42.83
$k = 100$	0.017	79.04	5.18	73.86

REFERENCES

Asmussen, S. 2008. *Applied Probability and Queues*, Volume 51. Springer New York, NY.

Heyman, D. P., and M. J. Sobel. 2004. *Stochastic Models in Operations Research: Stochastic Optimization*, Volume 2. Courier Corporation.

Kuntz, J., P. Thomas, G.-B. Stan, and M. Barahona. 2021. "Stationary Distributions Of Continuous-time Markov Chains: A Review Of Theory And Truncation-based Approximations". *SIAM Review* 63(1):3–64.

Seneta, E. 1980. "Computing The Stationary Distribution For Infinite Markov Chains". *Linear Algebra and Its Applications* 34:259–267.

Zheng, Z., A. Infanger, and P. W. Glynn. 2022. "Combining Numerical Linear Algebra with Simulation to Compute Stationary Distributions". In *2022 Winter Simulation Conference (WSC)*, 2342–2353. IEEE.

ACKNOWLEDGEMENT

We acknowledge excellent research assistance from Nanshan Jia with the numerical implementations. We thank the program committee members and chairs.

AUTHOR BIOGRAPHIES

PETER W. GLYNN is the Thomas Ford Professor in the Department of Management Science and Engineering at Stanford University. He is a Fellow of INFORMS and of the Institute of Mathematical Statistics, has been co-winner of Best Publication Awards from the INFORMS Simulation Society in 1993, 2008, and 2016, and was the co-winner of the John von Neumann Theory Prize from INFORMS in 2010. In 2012, he was elected to the National Academy of Engineering. In 2021, he received the INFORMS Simulation

Society's Lifetime Professional Achievement Award. His research interests lie in stochastic simulation, queueing theory, and statistical inference for stochastic processes. His email address is glynn@stanford.edu and his homepage is https://web.stanford.edu/∼glynn/.

ZEYU ZHENG is an assistant professor in the Department of Industrial Engineering & Operations Research at University of California Berkeley. He received his Ph.D. in Management Science and Engineering from Stanford University. He has done research in simulation and non-stationary stochastic modeling making. His email address is zyzheng@berkeley.edu and his homepage is http://zheng.ieor.berkeley.edu.