ABSTRACT

MOKASHI, ANUP CHANDRAKANT. The Simulation Start-up Problem: Performance Comparison of N-Skart and MSER-5. (Under the direction of James R. Wilson.)

The objective of this research is to conduct an extensive performance comparison of the batch-means procedures MSER-5 and N-Skart in terms of their effectiveness in handling the simulation start-up problem. Given a fixed-length simulation-generated time series that is potentially contaminated by a transient effect arising from the simulation’s initial condition, the deletion approach for the treatment of the simulation start-up problem is to determine a data-truncation point (or warm-up period) beyond which the observations have approximately steady-state behavior and thus can be used to compute point and confidence-interval (CI) estimators of the steady-state mean. MSER-5 uses the data-truncation point that minimizes the half-length of the usual batch-means CI based on batches whose size is always 5 observations. N-Skart applies a randomness test to spaced batch means in order to determine sufficiently large sizes for each batch and its preceding spacer such that beyond the initial spacer (which is taken to define the data-truncation point), the spaced batch means are approximately independent of each other and the simulation’s initial condition; then using truncated nonspaced batch means, N-Skart exploits separate adjustments to the CI half-length that account for the effects on the distribution of the underlying Student’s t-statistic arising from skewness and autocorrelation of the batch means. To compare both the methods in terms of the accuracy and reliability of their point and CI estimators of the steady-state mean, the experimental performance comparison used fixed-length simulation output sequences exhibiting patterns of initialization bias that are representative of many real-world simulation output processes. The results provide substantial evidence that the point estimator delivered by N-Skart generally has substantially smaller bias, variance, and mean-squared error than that delivered by MSER-5; moreover in all cases, N-Skart’s CI estimator substantially outperformed that of MSER-5.
The Simulation Start-up Problem: Performance Comparison of N-Skart and MSER-5

by
Anup Chandrakant Mokashi

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Industrial Engineering
Raleigh, North Carolina
2010

APPROVED BY:

______________________________  ________________________________
Peter Bloomfield                Russell E. King

______________________________
James R. Wilson
Chair of Advisory Committee
DEDICATION

To my grandparents, parents, and my brother Archit.
BIOGRAPHY

Anup Chandrakant Mokashi was born on October 14, 1986, in the city of Satara, India. He was raised by his parents, Mr. Chandrakant Y. Mokashi and Mrs. Shubhada C. Mokashi, along with his brother Archit.

In the fall of 2004, he entered the Bachelor of Engineering Program in the Pune University system. He graduated with a degree in mechanical engineering in 2008. In the fall of 2008, he began his graduate study in industrial engineering at North Carolina State University. He has also worked as a teaching assistant in the Edward P. Fitts Department of Industrial and Systems Engineering at North Carolina State University.
ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. James R. Wilson for introducing me to the field of stochastic simulation and for the time and effort that he has spent in helping me with my research. It was a privilege for me to work under his guidance. I would like to thank Dr. Ali Tafazzoli Yazdi for his help and contributions during my research. I would like to thank Dr. Russell E. King for serving on my advisory committee and for his help as the Director of Graduate Programs on various occasions during my graduate study. I would also like to thank Dr. Peter Bloomfield for serving on my advisory committee. Finally, I would like to thank my family for the love and support they have given me.
# TABLE OF CONTENTS

List of Tables ........................................................................................................ vi
List of Figures ........................................................................................................ vii

Chapter 1 Introduction .............................................................................................. 1
  1.1 Problem Statement and Research Objectives .................................................. 1
  1.2 Organization of the Thesis .............................................................................. 2

Chapter 2 Literature Review ...................................................................................... 4
  2.1 The Simulation Start-up Problem .................................................................... 4
  2.2 Review of Previous Literature on the Simulation Start-up Problem ............... 6
  2.3 Overview of N-Skart and MSER5 ................................................................. 14
    2.3.1 MSER5 ................................................................................................. 14
    2.3.2 N-Skart ............................................................................................... 16

Chapter 3 Performance Comparison of N-Skart and MSER-5 ................................. 20
  3.1 Measures for Comparison of N-Skart and MSER-5 ........................................ 20
  3.2 Description of Test Problems ....................................................................... 22
    3.2.1 $M/M/1$ Queue-Waiting-Time Process with Empty Initial Condition and 90% Utilization .......................................................... 22
    3.2.2 $M/M/1$ Queue-Waiting-Time Process with Empty Initial Condition and 80% Utilization .......................................................... 30
    3.2.3 $M/M/1$ Queue-Waiting-Time Process with 113 Initial Customers and 90% Utilization .......................................................... 37
    3.2.4 First-Order Autoregressive (AR(1)) Process ........................................ 45
    3.2.5 AR(1)-to-Pareto (ARTOP) Process ...................................................... 53

Chapter 4 Conclusions and Future Research ............................................................ 62
  4.1 Conclusions ................................................................................................. 62
  4.2 Contributions of the Current Research ......................................................... 64
  4.3 Directions for Future Research ..................................................................... 64

References .............................................................................................................. 66

Appendices .............................................................................................................. 68
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Performance of MSER-5 and N-Skart in the $M/M/1$ queue-waiting-time process with 90% server utilization and empty-and-idle initial condition</td>
<td>24</td>
</tr>
<tr>
<td>3.2</td>
<td>Performance of MSER-5 and N-Skart in the $M/M/1$ queue-waiting-time process with 80% server utilization and empty-and-idle initial condition</td>
<td>31</td>
</tr>
<tr>
<td>3.3</td>
<td>Performance of MSER-5 and N-Skart in the $M/M/1$ queue-waiting-time process with 90% server utilization and 113 initial customers</td>
<td>39</td>
</tr>
<tr>
<td>3.4</td>
<td>Performance of MSER-5 and N-Skart in the AR(1) process</td>
<td>47</td>
</tr>
<tr>
<td>3.5</td>
<td>Performance of MSER-5 and N-Skart in the ARTOP process</td>
<td>56</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

| Figure 2.1 | Transient and steady-state density functions for a particular stochastic process $X_1, X_2, \ldots$ and initial condition $X_0$ | 5 |
| Figure 2.2 | Flowchart of MSER-5 | 16 |
| Figure 2.3 | Flowchart of N-Skart | 19 |
| Figure 3.1 | A realization of the $M/M/1$ queue-waiting-time process with empty-and-idle initial condition and 90% server utilization | 23 |
| Figure 3.2 | Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0, \rho = 0.9, \text{ and } N = 10,000$ | 25 |
| Figure 3.3 | Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0, \rho = 0.9, \text{ and } N = 20,000$ | 26 |
| Figure 3.4 | Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0, \rho = 0.9, \text{ and } N = 50,000$ | 27 |
| Figure 3.5 | Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0, \rho = 0.9, \text{ and } N = 200,000$ | 28 |
| Figure 3.6 | A realization of the $M/M/1$ queue-waiting-time process with empty-and-idle initial condition and 80% server utilization | 30 |
| Figure 3.7 | Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0, \rho = 0.9, \text{ and } N = 10,000$ | 32 |
| Figure 3.8 | Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0, \rho = 0.8, \text{ and } N = 20,000$ | 33 |
| Figure 3.9 | Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0, \rho = 0.8, \text{ and } N = 50,000$ | 34 |
| Figure 3.10 | Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0, \rho = 0.8, \text{ and } N = 200,000$ | 35 |
| Figure 3.11 | A realization of the $M/M/1$ queue-waiting time process with 113 initial customers and 90% server utilization | 37 |
| Figure 3.12 | A depiction of the transient behavior of the $M/M/1$ queue-waiting time process with 113 initial customers and 90% server utilization for 3 independent replications | 38 |
| Figure 3.13 | Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with 113 initial customers, $\rho = 0.9, \text{ and } N = 10,000$ | 40 |
Figure 3.14 Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with 113 initial customers, $\rho = 0.9$, and $N = 20,000$.

Figure 3.15 Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with 113 initial customers, $\rho = 0.9$, and $N = 50,000$.

Figure 3.16 Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with 113 initial customers, $\rho = 0.9$, and $N = 200,000$.

Figure 3.17 A realization of the AR(1) Process (3.7) $X_0 = 0$, $\mu_X = 100$, and $\rho = 0.995$.

Figure 3.18 A depiction of the transient behavior of the AR(1) Process (3.7) $X_0 = 0$, $\mu_X = 100$, and $\rho = 0.995$ for 3 independent replications.

Figure 3.19 Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to AR(1) process (3.7) with $X_0 = 0$, $\mu_X = 100$, $\rho = 0.995$, and $N = 10,000$.

Figure 3.20 Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to AR(1) process (3.7) with $X_0 = 0$, $\mu_X = 100$, $\rho = 0.995$, and $N = 20,000$.

Figure 3.21 Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to AR(1) process (3.7) with $X_0 = 0$, $\mu_X = 100$, $\rho = 0.995$, and $N = 50,000$.

Figure 3.22 Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to AR(1) process (3.7) with $X_0 = 0$, $\mu_X = 100$, $\rho = 0.995$, and $N = 200,000$.

Figure 3.23 A realization of the ARTOP Process.

Figure 3.24 A depiction of the transient behavior of the ARTOP Process for 3 independent replications.

Figure 3.25 Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to ARTOP process (3.12) with $N = 10,000$.

Figure 3.26 Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to ARTOP process (3.12) with $N = 20,000$.

Figure 3.27 Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to ARTOP process (3.12) with $N = 50,000$.

Figure 3.28 Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to ARTOP process (3.12) with $N = 200,000$.

Figure 1 C Code for N-Skart.

Figure 2 C Code for MSER-5.
Chapter 1

Introduction

Simulation studies have become an important tool in the modeling and analysis of real-world systems. Typically, we are interested in knowing about the characteristics of a dynamic stochastic system in its long-run steady-state operational condition. Steady-state (nonterminating) simulation, unlike a terminating simulation, does not have any specified starting or stopping conditions. As a result, the parameters of interest that are to be estimated are defined over time as the time horizon tends to infinity. While designing nonterminating simulation experiments, an arbitrary starting condition is chosen and the simulation is run for a sufficient number of output responses so as to approximate the long-run average behavior of the system. Ideally, the starting condition should not affect the estimates of the parameters of interest. However, the starting condition introduces a transient in the simulation output responses which results in biased estimates of the steady-state parameters of interest. The problem of initialization bias has been a long-standing problem in the field of simulation modeling and analysis and has been the subject of several studies conducted in the past.

1.1. Problem Statement and Research Objectives

In this study, the problem of initialization bias is described in detail, and an overview is provided of the various approaches used for the treatment of initialization bias in simulation output processes. Special emphasis is given to the truncation approach, and a detailed explanation is presented for the intuition behind this approach. Various methods employing the truncation rationale are described, and two of the methods proposed in recent times are selected for comparison—namely, N-Skart (Tafazzoli, Steiger, and Wilson 2010) and MSER-5 (Franklin and White 2008).

N-Skart (Tafazzoli, Steiger, and Wilson 2010) is a nonsequential procedure designed to deliver a confidence interval (CI) for the steady-state mean of a simulation output process when a single simulation-generated time series of arbitrary size is supplied by the user, and a required coverage probability for the CI is specified. N-Skart applies a randomness test to spaced batch means in order to...
determine sufficiently large sizes for each batch and its preceding spacer such that beyond the initial spacer (which is taken to define the data-truncation point), the spaced batch means are approximately independent of each other and the simulation’s initial condition; then using truncated, nonspaced batch means, N-Skart makes separate adjustments to the half-length of the CI in order to account for the effect of skewness and autocorrelation of the batch means on the underlying Student’s $t$-statistic. If the sample size is large enough, N-Skart delivers a point estimate and a CI for the steady-state mean of the process that is approximately free of initialization bias.

MSER-5 (Franklin and White 2008) is a variant of the MSER procedure first proposed by White (1997). It is a truncation heuristic for resolving the start-up problem which has received a considerable amount of attention in recent times due to its simplicity. Given a finite sequence, MSER-5 first computes batch means from adjacent (nonoverlapping) batches, each consisting of five observations; then MSER-5 computes the usual CI for the steady-state mean based on the assumption that the batch means are randomly sampled from a normal distribution; and finally MSER-5 sequentially recomputes the batch means CI after deleting (truncating) progressively more initial batch means until the width of the marginal confidence interval about the truncated sample mean is minimized. The truncated sequence can then be considered to be approximately free of initialization bias; and the corresponding truncated sample average of the batch-means is supposed to have minimal mean-squared error as an estimator of the steady-state mean. Moreover, the usual batch means CI for the steady-state mean computed from the batch means beyond MSER-5’s “optimal” truncation point is supposed to be a valid CI—that is, its actual coverage probability should be (nearly) equal to the user-specified nominal coverage probability.

MSER-5 and N-Skart are compared in terms of their effectiveness in removing the initial transient for a carefully selected set of test problems with patterns of initialization bias that are typical of many large-scale simulation applications. Four different sample sizes are used to test the effectiveness of the methods for large, medium, and relatively small output streams. The performance of the two methods is analyzed both in terms of the following:

(a) Statistics that are indicative of the residual initialization bias after truncation; and

(b) Graphs of those statistics.

Consistency of the performance of both the methods is tested by performing 1,000 independent replications for both N-Skart and MSER-5 on 1,000 independent realizations of each test process.

1.2. Organization of the Thesis

The remainder of the thesis is organized as follows: Chapter 2 presents a brief overview of the simulation start-up problem and the previous literature on this topic. Chapter 3 describes the measures of
performance for the comparison of the two methods along with the experimental design of the test problems selected for the comparison. Chapter 4 presents the results of the experiments. Chapter 5 contains the concluding remarks and suggestions for future research.
Chapter 2

Literature Review

2.1. The Simulation Start-up Problem

Simulation studies are primarily conducted in order to obtain reliable information about parameters of interest, which cannot be readily obtained through analytical methods. For the analysis of simulation output, we use several statistical methods. Most of these statistical methods are applied under the assumptions that the simulation output observations are independent and identically distributed. However, many real world processes and simulations are nonstationary and correlated. Consider a stochastic output process from a single simulation run given by $X_1, X_2, X_3, \ldots$. In general, the process can be expressed as $\{X_j\}$ for $j = 1, 2, 3, \ldots$. These successive observations in this process, in general, will neither be independent nor identically distributed; moreover in many simulation-generated responses are not even approximately distributed according to a normal distribution. As a result, conventional statistical methods (such as the usual confidence interval based on Student’s $t$-distribution) cannot be applied directly.

For above defined simulation output process, let the initial condition at the start of the simulation be represented by $X_0$. Let $F_j(x|X_0) = P(X_j \leq x|X_0)$, where $F_j(x|X_0)$ is the transient distribution of the process at time $j$, for initial condition $X_0$. The transient distribution varies for different values of $j$ and $X_0$. If $F_j(x|X_0) \rightarrow F(x)$ as $j \rightarrow \infty$ for all $x$ and for any initial conditions $X_0$, then $F(x)$ is called the steady-state distribution of the output process $X_1, X_2, X_3, \ldots$. Ideally, $F_j(x|X_0) \rightarrow F(x)$ only as $j \rightarrow \infty$; but in practice, there will be a finite time index beyond which these distributions would be approximately the same as each other. Thus, if $d$ is such a point which satisfies this condition, then each observation $X_j$ with index $j > d$ can be said to have been sampled approximately from the steady-state distribution. Also, the individual observations $\{X_j : j = d + 1, d + 2, \ldots\}$ would not be independent, but are assumed to constitute an approximately covariance stationary process. The steady-state distribution $F(x)$ is independent of the initial condition $X_0$, but the rate of convergence of $F_j(x|X_0)$ to $F(x)$ is not. Also, the steady-state distribution is not necessarily a normal distribution. Figure 2.1 illustrates the
convergence of the transient probability distributions of the random variable \( \{X_j\} \) to the steady-state distribution as \( j \to \infty \).

Figure 2.1: Transient and steady-state density functions for a particular stochastic process \( X_1, X_2, \ldots \) and initial condition \( X_0 \)

The simulation for a particular system may be terminating or nonterminating depending on the objectives of the study. A terminating simulation is one for which a specified event \( E \) determines the length of each simulation run. A nonterminating simulation does not have any such event to determine the end of a replication. The measures of performance of a terminating or nonterminating simulation are greatly influenced by the state of the system at the beginning of the simulation run, represented by \( X_0 \). Suppose we are interested in determining a steady-state parameter \( \phi \); i.e. a parameter of the steady-state distribution \( F(x) \) of a nonterminating simulation. One practical difficulty in estimating \( \phi \) is that we are trying to estimate the parameters of a distribution \( F(x) \) which would be obtained only as \( j \to \infty \), but in real-world simulations, we only have a finite set of \( n \) observations from which we try to estimate the
steady-state parameter of interest. Also, it is not always possible to choose the initial condition $X_0$ that is representative of the steady-state behavior of the system. Thus, if we are trying to estimate

$$\phi = \mathbb{E}(X) = \int_{-\infty}^{+\infty} x dF(x),$$

(2.1)

i.e., the steady state mean of the process, then the sample mean

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

(2.2)

will be a biased estimator of $\phi$ for all values of $n$. This problem is called as the initialization-bias problem or the simulation start-up problem.

### 2.2. Review of Previous Literature on the Simulation Start-up Problem

Conway (1963) discussed the artificiality introduced in a simulation output due to its abrupt starting, unlike the real world system it represents, and the resulting deviation from equilibrium conditions in steady-state operation. He elaborated the effect of initial conditions on the stochastic dependence between observations in both steady-state and transient simulations. He also suggested truncation as an efficient method of removing the effect of initial conditions and proposed a rule for truncation wherein the initial set of observations were deleted until the first of the residual series was neither the maximum nor the minimum. Another approach suggested by him was to start the simulation using initial conditions that are representative of the steady state. This would help to shorten, if not eliminate, the time required for a simulation to reach equilibrium conditions.

Fishman (1972) investigated the effect of the initial transient in a simulation output on the steady-state mean of a process, using the first-order autoregressive process as a test problem. Fishman proposed that the bias in the simulation output is directly proportional to the deviation of the initial condition from the steady-state parameter of interest and inversely proportional to the run length of the simulation. He also investigated the effect of truncating an initial set of observations on the bias and variance of the simulation output. In the results presented, he questioned the effectiveness of the truncation method against the resulting increase in variance in the truncated output sequence.

Welch (1983) defined the start-up problem as follows: with respect to a particular estimand, the problem is to identify a truncation point beyond which the expected value of the estimate of the parameter of interest is approximately equal to its limiting value. He proposed a graphical solution to the start-up problem which requires the user to perform multiple replications of the system. The mean and the variance at any point in simulated time can be estimated by averaging the corresponding values across all replications. On the $i^{th}$ independent replication of the simulation, for $i = 1, \ldots, k$, let $X_{ij}$ denote the $j^{th}$ observation of the target output process for $j = 1, \ldots, n$, where $X_{00} = X_0$ is the common
initial condition for each replication. An estimate of the transient mean function

$$\mu_j(X_0) = \mathbb{E}[X_{ij} | X_0] = \int_{-\infty}^{+\infty} xdF_j(x | X_0) \text{ for } j = 1, \ldots, n,$$

(2.3)
is the sample average transient mean function

$$\hat{\mu}_j(X_0) = \frac{1}{k} \sum_{i=1}^{k} X_{ij} \text{ for } j = 1, \ldots, n,$$

(2.4)

which is the sample mean response computed at the time index $j$, averaged across all $k$ replications. Similarly, an estimate of the transient variance function

$$\sigma_j^2(X_0) = \text{Var}[X_{ij} | X_0] = \int_{-\infty}^{+\infty} \{x - \mathbb{E}[X_{ij} | X_0]\}^2 dF_j(x) \text{ for } j = 1, \ldots, n$$

(2.5)
is the sample variance

$$\hat{\sigma}_j^2(X_0) = \frac{1}{k-1} \sum_{i=1}^{k} [X_{ij} - \hat{\mu}_j(X_0)]^2 \text{ for } j = 1, \ldots, n,$$

(2.6)
of all the responses observed at time $j$, computed across all $k$ replications. Thus, an approximate $100(1 - \alpha)\%$ confidence interval for $\mu_j(X_0)$ is

$$\hat{\mu}_j(X_0) \pm t_{1 - \alpha / 2, k-1} \frac{\hat{\sigma}_j(X_0)}{\sqrt{k}},$$

(2.7)

where $t_{1 - \alpha / 2, k-1}$ is the $(1 - \alpha)$ quantile of Student’s $t$-distribution with $k - 1$ degrees of freedom. Welch proposed identifying the end $d$ of the warm-up period (that is, the truncation point) by visual inspection of the sample average transient mean function (2.4) or of the confidence band (2.7) so that for the time index $j \geq d$, (2.4) or (2.7) have “settled down” and thus, approximately represent steady-state behavior. In this case, the truncated sample mean on replication $i$

$$\overline{X}_i(n,d) = \frac{1}{n-d} \sum_{j=d+1}^{n} X_{ij} \text{ for } i = 1, \ldots, k,$$

(2.8)
is an approximately unbiased estimator of the steady-state mean $\mu_X = \int_{-\infty}^{+\infty} xdF(x)$; and a $100(1 - \alpha)\%$ confidence interval for $\mu_X$ is

$$\overline{X}(k,n,d) \pm t_{1 - \alpha / 2, k-1} \frac{S_X(n,d)}{\sqrt{k}},$$

(2.9)

where

$$\overline{X}(k,n,d) = \frac{1}{k} \sum_{i=1}^{k} \overline{X}_i(n,d)$$

(2.10)
and

\[ S^2_{\overline{X}_{(n,d)}} = \frac{1}{k-1} \sum_{i=1}^{k} \left( \overline{X}_i(n,d) - \overline{X}(k,n,d) \right)^2 \]  \hspace{1cm} (2.11)

are the grand mean and the sample variance of the truncated sample means, respectively. To smooth out the short-term fluctuations in the sequence, Welch recommended the use of moving averages which would reduce the sensitivity of the simulation output data to these irregularities.

Gafarian, Ancker, and Morisaku (1977) proposed that a random variable was to be used to estimate the true truncation point. They suggested the following criteria for evaluating performance of various truncation rules:

- **Accuracy** — The ratio of expected value of the truncation point to the true truncation point should be close to 1;
- **Precision** — The coefficient of variation of the estimate of the truncation point should be close to 0;
- **Generality** — The truncation rule should perform well for a broad range of systems;
- **Simplicity** — The rule should be easy to implement for average practitioners; and
- **Cost** — Expenses in computer time required should be minimum.

Wilson and Pritsker (1978a, 1978b) presented a survey of the various start-up policies in practice which define the initial conditions and the truncation point in a simulation output process. They classified the policies into three broad categories as follows.

- **time series models**;
- **queueing models**; and
- **heuristic rules of thumb**.

They provided an overview of the various statistics used to evaluate start-up policies and stressed the importance of developing appropriate performance measures in order to compare the alternative start-up policies in a consistent manner. The suggestion put forward by them with respect to truncation rules was to develop an evaluation procedure for all the truncation methods that would focus on the behavior of the truncated sample mean and would consider the random variation in the truncation point. The evaluation procedure should also be able to characterize the random and systematic components of the resulting bias in the estimate of the truncated sample mean.
The resulting evaluation procedure developed by them used the bias, variance and mean-squared error of the truncated sample mean along with the confidence interval coverage for performance comparison of start-up policies which were defined as a combination of initial condition rules \((IC_i)\) and truncation rules \((TR_l)\). Given a simulation output process denoted by \(\{X_j : j = 1, \ldots, n\}\), the various initial condition rules and truncation rules used in their evaluation procedure are as follows.

Initial Condition rules:

- \(IC_1\) \(
\rightarrow\) Start the system “empty and idle”.

- \(IC_2\) \(
\rightarrow\) Set the initial condition \(X_0\) as close to the steady-state mode as possible.

- \(IC_3\) \(
\rightarrow\) Set the initial condition \(X_0\) as close to the steady-state mean as possible.

Truncation rules:

- \(TR_0\) \(
\rightarrow\) Retain all data.

- \(TR_1\) \(
\rightarrow\) Set the truncation point \(d\) such that the observation \(X_{d+1}\) in the given sequence is neither maximum nor minimum of the remaining sequence \(\{X_j : j = d + 1, \ldots, n\}\).

- \(TR_2\) \(
\rightarrow\) Set the truncation point as \(d = n\) when the simulation output sequence \(\{X_j\}\) has crossed its mean \(\bar{X}_n\) at least \(k - 1\) times i.e. \(\{\text{sgn}(X_t - \bar{X}_n) : 1 \leq t \leq n\}\) contains \(k\) runs.

- \(TR_3\) \(
\rightarrow\) Set \(d = n\) when the batch-means of \(k\) most recent batches of size \(b\) fall within an interval of length \(\epsilon\).

The steps in the evaluation procedure can be summarized as follows.

1. Compute the bias, variance and mean-squared error for all the start-up policies of interest \((IC_i, TR_l)\).

2. Estimate the distribution of the truncation point \(d\) over independent simulation runs for each of the start-up policies.

3. Calculate the averaged values of the bias, variance and mean-squared-error.

4. Select a base policy. Calculate the half-length of a nominal 100 \((1 - \alpha\%)\) confidence interval (CI) for the steady state mean \(\phi\) based on individual replications. Obtain the standard average half-length of the CI for the base policy. Adjust the nominal confidence level for all other policies so that the half-length of the resulting confidence intervals is the same as for the policy.

5. Compute coverage probabilities for all possible values of the truncation point \(d\) for individual replications of each start-up policy. Averaged CI coverages \(\bar{C}_n(i, l)\) are calculated for each start-up policy \((IC_i, TR_l)\). Since all policies have been normalized with respect to the base policy, the alternatives can be compared in terms of average CI coverages.
On applying the above comprehensive evaluation procedure to a single-server queue ($M/M/1/15$ queue with arrival rate $\lambda = 4.5$ and service rate $\mu = 5$) and a machine repair system ($M/M/3/14/14$ queue with failure rate $\lambda = 0.2$ and repair rate $\mu = 0.5$), it was observed that specifying an initial condition close to the steady-state mode was optimal. This is because departure from this condition resulted in increased variance and reduced CI coverage. Also, with respect to improving the estimate of the steady-state mean, selection of initial condition played a more crucial role than truncation because the truncation rules under study were extremely sensitive to parameter misspecification which resulted in considerable loss of CI coverage. It was also observed that incorporating prior information about the process under study into the evaluation procedure significantly reduced the variability in the results.

Law (1983) presented a comprehensive overview of the various statistical analysis techniques for simulation output data. Specifically, for steady-state simulation analysis, he categorized the output analysis techniques as fixed-sample-size procedures or sequential procedures. He further classified the fixed-sample-size procedures as:

- those that seek independent observations;
- those that seek to estimate dependence among output responses;
- those that exploit special structure of the underlying process; and
- those based on standardized time series.

Law (1983) also presented an in-depth analysis of the various techniques employed for mitigation of the start-up problem. He evaluated the deletion approach with respect to the following criteria:

- covariance stationarity;
- point-estimator quality; and
- confidence-interval quality.

A brief description of the same is presented below.

Output analysis techniques such as batch means, autoregressive representation, and spectrum analysis, which are typically employed when we conduct a single long run of the simulation, assume that the underlying process is covariance stationary. If our goal is to apply the above techniques to a covariance stationary process, then the truncation rationale used should ensure that the residual observations in the truncated sequence are approximately covariance stationary.

Another approach for evaluating the efficiency of truncation is to consider the properties of the point estimator for some steady-state performance measure of interest such as the steady-state mean; and then we are interested in the bias and variance of the truncated sample mean as an estimator of the steady-state mean. Law (1983) made the observation that truncation provided improved point-estimators of
the steady-state mean for most processes but also cautioned regarding exceptions to this trend. When
considering the variance, if observations at the beginning of the output sequence have particularly large
variances, then truncation could decrease the variance of the estimator. Truncation also reduces the
mean-squared error when the initialization bias is high and the observations are highly correlated. Also,
as the overall sample size increases, generally the ratio of the truncation point to the overall run length
(sample size) decreases.

The third approach for determining the efficiency of truncation is to assess its impact on the confidence-
interval quality. For a single long run of the simulation, the truncation approach does not have any
significant impact when the sample size is moderate and the methods employed for constructing the
confidence interval are batch means, autoregressive representation, or spectrum analysis. If the number
of truncated observations is very large relative to the actual sample size, then truncation could actually
result in a degradation in the confidence-interval coverage. As against this, Law (1983) showed that
replication and deletion, when used in conjunction, provided improved results. For a fixed sample size,
truncation increased the expected value of the confidence-interval half length. Also, truncation generally
decreases the replication length required.

Kelton (1989) investigated the feasible methods for initializing simulations that would lead to lower
estimator bias or less requisite deletion. He compared deterministic and stochastic initialization rules
and suggested forms for the initial distribution using the maximum entropy principle. His work can be
summarized as follows. The initialization states of steady-state simulations can be broadly classified as
deterministic or stochastic. Starting each simulation run in an identical deterministic state is the most
common method of initializing replications. Each replication, when initialized by the above method,
will pass through a transient phase that is identical across replications, since the initial state is always
the same; however, it must be recognized that on independent replications of the simulation model
with the same starting state, the observed responses will exhibit different patterns of variation about the
transient mean function defined in equation (2.3). Therefore even when the same deterministic starting
date is used on each independent replication of the simulation, different realizations of the target output
process that are conditionally independent of each other, given the simulation’s starting state.

Alternatively, one can draw the initial state from some probability distribution instead of specifying
it to be the same deterministic value for all replications. Thus, we can allow for different re-
alized initialization states across replications and yet retain the probabilistic identity of the replications.
In the absence of sufficient information regarding the form of the initial distribution, one can
utilize the principle of maximum entropy which produces a discrete probability mass function (p.m.f.)
\{p(x_1), p(x_2), \ldots, p(x_k)\} of a random variable \(X\) on the finite domain \(\{x_1, \ldots, x_k\}\) that maximizes the
p.m.f.’s entropy

\[
H(p) = -E[\ln(X)] = - \sum_{i=1}^{k} p(x_i)\ln p(x_i) \tag{2.12}
\]
subject to the $t$ expectation constraints

$$\sum_{i=1}^{k} g_j(x_i)p(x_i) = v_j \text{ for } 1 \leq j \leq t.$$  \hspace{1cm} (2.13)

where
g_j \rightarrow \text{specified constants}
g_i \rightarrow \text{given real-valued functions}

Thus, we can express the problem as

$$\max_{p} H(p) \text{ subject to } \sum_{i=1}^{k} p(x_i) = 1 \text{ and } (2.13)$$  \hspace{1cm} (2.14)

The p.m.f. resulting from entropy maximization is the ”maximally noncommittal” distribution that obeys the constraints. In the absence of any information regarding the expected system state in steady state, a discrete uniform distribution can be specified whose limits can be determined through a set of initial pilot runs. Kelton used the following initialization methods for comparison:

- the batch means method for a single long run of the simulation;
- the replication method without deletion of the warm-up period;
- the replication method with deletion of the warm-up period;
- uniform initial distribution without deletion of the warm-up period;
- uniform initial distribution with deletion of the warm-up period;
- geometric initial distribution without deletion of the warm-up period; and
- geometric initial distribution without deletion of the warm-up period.

The evaluation criteria for the different methods were as follows.

- plots of the expected transient response as a function of discrete time;
- the percent bias; and
- the time required to attain near-steady state.

Kelton concluded that the stochastic initialization methods performed considerably better than the replication and batch-means methods with respect to reducing the bias in the simulation output and also helped in reducing the amount of subsequent truncation required to get a simulation output that is approximately free of the initial transient. Also, the 90\% CI coverages provided by stochastic initialization
methods are considerably higher than the other methods. The primary drawback of stochastic initialization is that in complex, large-scale simulations, it is exceedingly difficult if not impossible to do the following:

(a) construct the relevant joint distribution of all the system state variables that define the simulation’s initial state; and then

(b) randomly sample from the joint distribution in (a).

Starting the simulation in some convenient starting state and then clearing statistics at the end of a warm-up period of appropriate length—that is, using data truncation to solve the simulation start-up problem—may be viewed as one method for stochastic initialization, because the system status at the end of the warm-up period may be regarded as a probabilistically assigned initial condition for the simulation-generated output responses that are recorded beyond the end of the warm-up period.

Grassmann (2008) questioned the utility of the truncation approach for determining the steady state of simulation output process. He proposed that if one starts a simulation with an initial state that has a high equilibrium probability, there should not be a warm-up period. He compared different starting states for a number of systems and calculated the mean-squared error (MSE) for these different initial states using the Markov modeling approach and the Markov event system (MES) as a modeling tool. Grassmann used the following test problems to illustrate his conclusions:

- a one-server queue;
- multiserver queues; and
- tandem queues.

He observed that starting the system in with different initial states did not affect the MSE significantly. Specifically, for the one-server queue, selecting the starting value according to the equilibrium probabilities yielded the largest MSE compared with a few other fixed starting states. He further noted that if the system under consideration regenerates (i.e. the same stochastic initial state is revisited), then the utility of the truncation approach is questionable since the warm-up period at the beginning of each regeneration cycle would be required to be truncated.

Grassmann also observed that in simulation outputs, the difference between variance and the MSE becomes negligible as the sample size increases and the variance is strongly influenced by extreme values. He questioned the utility of the MSE as a performance measure and suggested that it be replaced by an alternative measure such as the mean absolute deviation.

The main difficulty with Grassmann’s recommendation to use a fixed starting state with a relatively large steady-state probability is related to the problem of implementing Kelton’s recommendation to use a randomly sampled starting state—in complex, large-scale simulations, both recommendations are extremely difficult to implement in practice.
2.3. Overview of N-Skart and MSER5

2.3.1. MSER5

MSER-5 (Franklin and White 2008) is a modification of the Marginal Confidence Rule (MCR) or the Marginal Standard Error Rule (MSER) proposed by White (1997). There are two key notions about the intuition behind MSER-5:

- MSER-5 optimizes the objective function that we care the most about in simulation studies—i.e., the confidence interval (CI) for the steady-state mean $\mu$; and
- MSER-5 provides a reasonable method for determining the approximate end of the warm-up period.

MSER-5 aims at balancing improved accuracy achieved through reduction in the bias of the estimate of the steady-state mean, with decreased precision due to reduction in sample size. It follows the rationale that given an output sequence $\{X_1, X_2, \ldots, X_N\}$ of length (sample size) $N$, the observations later in the sequence provide a more accurate estimate of the central tendency at steady state; and hence, the initial observations, which are suspected to be further from the steady-state mean, should be truncated. The heuristic works backwards in time gathering more and more observations. As long as these observations are representative of the steady state, the estimate of the width of the confidence interval around the estimate of the steady-state mean will continue to decrease. As soon as the observations collected are not representative of the steady state behavior, the width of the confidence interval would start to increase. Thus, the initial observations in the output sequence should be truncated to the extent that deleting those observations minimizes the length of the confidence interval for the steady-state mean based on the remaining output sequence. The improvement of MSER-5 over MCR is that it batches together 5 observations in the output sequence in order to ensure better behavior in the confidence interval. Therefore, MSER5 uses as its basic data items, the batch means computed from the original process $\{X_i : i = 1, \ldots, N\}$ with batch size 5,

$$Z_j = \frac{1}{5} \sum_{i=5(j-1)+1}^{5j} X_i \quad \text{for} \quad j = 1, \ldots, k = \left\lfloor \frac{N}{5} \right\rfloor.$$  \hspace{1cm} (2.15)

In terms of the batch means $\{Z_j : j = 1, \ldots, k\}$, MSER-5 determines $d^*$, the optimal number of batches in the warm-up period, as follows:

$$d^* = \arg\min_{0 \leq d < k-1} \frac{z_{1-\alpha/2}S_{Z_k}(k,d)}{\sqrt{k-d}},$$  \hspace{1cm} (2.16)

where:

- $z_{1-\alpha/2}$ denotes the $(1 - \alpha/2)$ quantile of the standard normal distribution; and
- $Z_k(k,d)$ denotes the truncated sample mean of the batch means with truncation point $d$,
\[
Z(k, d) = \frac{1}{k-d} \sum_{j=d+1}^{k} Z_j;
\]
(2.17)

and \(S_Z^2(k, d)\) denotes the corresponding truncated sample variance,

\[
S_Z^2(k, d) = \frac{1}{k-d} \sum_{j=d+1}^{k} (Z_j - Z(k, d))^2.
\]
(2.18)

The MSER-5 truncation criterion can then be expressed as

\[
\text{MSER5}(k, d) = \frac{S_Z^2(k, d)}{k-d};
\]
(2.19)

and the truncation point \(d\) is set to minimize \(\text{MSER5}(k, d)\) for \(0 \leq d < k\). The final point estimator of the steady-state mean delivered by MSER-5 is the truncated sample mean \(\overline{Z}(k, d^*)\); and the associated nominal 100(1 − \(\alpha\))% confidence interval for \(\mu\) is

\[
\overline{Z}(k, d^*) \pm \frac{z_{1-\alpha/2} S_Z(k, d^*)}{\sqrt{k-d^*}}.
\]
(2.20)

It has been observed that MSER-5 can sometimes deliver a truncation point at the end of the data series—that is, \(d^* = k - 2\). According to Delaney (1995) and Spratt (1998), this is because the method can be overly sensitive to observations at the end of the data series that are close in value. Spratt (1998) proposed that this error can be avoided, partially, by not allowing the algorithm to consider the standard error calculated from the last few data points. Hence, while calculating the MSER-5 truncation criterion, instead of using a minimum sample of 2 batches to calculate the truncated sample variance \(S_Z^2(k, d)\), we choose an arbitrarily greater sample size. Thus, we can summarize MSER-5 as a sequential procedure which utilizes a modified version of the nonoverlapping batch means procedure in order to deliver a truncation point and a mean estimate with a 100(1 − \(\alpha\))% CI for a simulation output sequence.

In Chapter 3, we evaluate the performance of MSER-5 in terms of the point and CI estimators of \(\mu_X\) for a wide range of test processes. We consider the results obtained from the original N-Skart algorithm as well as the results obtained from the modified algorithm with the correction proposed by Delaney and Spratt. For the modified algorithm, we consider a minimum of 6 batches for calculating the sample variance \(S_Z^2(k, d)\). Thus, in the modified version of MSER-5, the truncation point is defined by

\[
d^* = \arg\min_{0 \leq d \leq k-6} \frac{S_Z^2(k, d)}{k-d}
\]
(2.21)

The steps for implementing MSER-5 have been summarized in the following figure.
Flowchart of MSER-5

2.3.2. N-Skart

N-Skart (Tafazzoli 2009; Tafazzoli, Steiger, and Wilson 2010) is a nonsequential procedure that is designed to deliver both point and confidence-interval estimates of the steady-state mean of a simulation output process that are approximately free of initialization bias. It can be considered as an extension of the classical method of nonoverlapping batch means (NBM). The key notions about the intuition behind N-Skart can be summarized as follows.

- N-Skart provides an accurate point estimator of the steady-state mean that is approximately free of initialization bias;
- N-Skart provides a sufficiently stable estimator of the standard error of the point estimator that accounts for correlation among simulation responses used to compute the point estimator; and
- N-Skart provides a suitable adjustment to the critical value of the Student’s $t$-distribution that accounts for nonnormality (skewness) in the data used to compute the point estimator and its
Input to N-Skart is in the form of a single simulation-generated time series \( \{X_i : i = 1, \ldots, N\} \) of arbitrary size \( N \); and the user specifies the required coverage probability \( 1 - \alpha \) (where \( 0 < \alpha < 1 \)) for a confidence interval, based on the data set. N-Skart addresses the problem of the initial transient by successively applying the randomness test of von-Neumann (1941) to spaced batch means with progressively increasing sizes for each batch and its preceding spacer until the spaced batch means are finally determined to be approximately independent of each other and the simulation’s initial condition. When the randomness test is passed, the observations in the initial spacer are deleted (this is the warm-up period). The truncated sequence is then used to compute point and CI estimates of \( \mu_X \) and CI.

N-Skart also tackles the nonnormality problem by using a modified Cornish-Fisher expansion (Johnson 1978, Willink 2005) for the classical batch-means Student’s \( t \)-ratio that also incorporates a term which accounts for any skewness in the set of truncated, nonspaced batch means. The correlation problem is addressed by using an autoregressive approximation to the autocorrelation function of the truncated, nonspaced batch means.

N-Skart delivers an point estimate and a skewness- and autocorrelation-adjusted confidence interval for the true mean based on the truncated sequence. The detailed steps for implementing N-Skart are described below and summarized in the following figure. Given the simulation-generated time series \( \{X_i : i = 1, \ldots, N\} \) of length \( N \), N-Skart handles the start-up problem by applying the randomness test of von-Neumann (1941) to determine sufficiently large values of the batch size \( m \) and spacer size \( dm \) (where \( m \geq 1 \) and \( d \geq 0 \)) such that a set of \( k \) spaced batch means

\[
Y_j(m, d) = \frac{1}{m} \sum_{i=[j(d+1)-1]m+1}^{j(d+1)m} X_i \quad \text{for} \quad j = 1, \ldots, k
\]  

(2.22)

are approximately independent of each other and of the initial condition \( X_0 \). Because the spacer preceding the \( j \)th batch of size \( m \) consists of the ignored (deleted) observations

\[
\{X_i : i = (j-1)(d+1)m + 1, \ldots, [j(d+1) - 1]m\},
\]  

(2.23)

we see in particular that the initial spacer (that is, the spacer obtained by taking \( j = 1 \) in (2.23)) consists of the observations

\[
\{X_i : i = 1, \ldots, dm\}
\]  

(2.24)

so that the first spaced batch mean

\[
Y_1(m, d) = \frac{1}{m} \sum_{i=dm+1}^{(d+1)m} X_i
\]  

(2.25)
is approximately independent of the initial condition \(X_0\); moreover all the spaced batch means \(\{Y_j(m,d) : j = 1, \ldots, k\}\) are approximately independent of each other and the initial condition \(X_0\). It follows that any effects due to initialization bias are limited to the initial spacer (2.24); and this is the reason why N-Skart uses the initial spacer (2.24) as the warm-up period so that the first \(dm\) observations are truncated.

Beyond the data truncation point \(dm\), N-Skart next computes the \(k'\) truncated, nonspaced batch means with batch size \(m\)

\[
Y_j(m) = \frac{1}{m} \sum_{i=(d+j-1)m+1}^{(d+j)m} X_i \quad \text{for} \quad j = 1, \ldots, k',
\]  

(2.26)

where \(k'\) is taken large enough to use the entire data set \(\{X_i : i = 1, \ldots, N\}\); and then N-Skart computes the sample mean and variance of the truncated, nonspaced batch means,

\[
\bar{Y}(m,k') = \frac{1}{k'} \sum_{j=1}^{k'} Y_j(m) \quad \text{and} \quad S_{m,k'}^2 = \frac{1}{k'-1} \sum_{j=1}^{k'} [Y_j(m) - \bar{Y}(m,k')]^2
\]  

(2.27)

respectively. Finally, N-Skart delivers an asymptotically valid \((1 - \alpha)\)% skewness-and autocorrelation-adjusted CI for \(\mu_X\) having the form

\[
\left[ \bar{Y}(m,k') - G(L) \frac{AS_{m,k'}^2}{k'}, \bar{Y}(m,k') - G(R) \frac{AS_{m,k'}^2}{k'} \right],
\]  

(2.28)

where the skewness adjustments \(G(L)\) and \(G(R)\) are defined in terms of the function

\[
G(\zeta) = \sqrt{1 + 6\beta(\zeta - \hat{\beta})} - 1 \quad \text{for all real} \quad \zeta \quad \text{where} \quad \hat{\beta} = \frac{\hat{\beta}_{m,k''}}{6\sqrt{k''}},
\]  

(2.29)

and \(\hat{\beta}_{m,k''}\) is an approximately unbiased estimator of the marginal skewness of the \(k''\) spaced batch means of the form (2.22) that can be computed from the entire data set \(\{X_i : i = 1, \ldots, N\}\) of size \(N\). The skewness-adjustment function \(G(\cdot)\) has the arguments

\[
L = t_{1 - \alpha/2,k''-1} \quad \text{and} \quad R = t_{\alpha/2,k''-1}.
\]  

(2.30)

The correlation adjustment \(A\) is computed as

\[
A = \left[ 1 + \hat{\phi}_{Y(m)} \right] / \left[ 1 - \hat{\phi}_{Y(m)} \right],
\]  

(2.31)

where the standard estimator of the lag-one correlation of the truncated, nonspaced batch means (2.26) is
\[ \hat{\varphi}_Y(m) = \frac{1}{k' - 1} \sum_{j=1}^{k'-1} \left[ Y_j(m) - \bar{Y}(m, k') \right] \left[ Y_{j+1}(m) - \bar{Y}(m, k') \right] \frac{S^2_{m,k'}}{S^2_{m,k'}}. \] (2.32)

(Note that in (2.29), the indicated cube root \( \sqrt[3]{1 + 6\beta(\zeta - \beta)} \) is understood to have the same sign as the quantity \( 1 + 6\beta(\zeta - \beta) \).)

From (2.28), (2.29) and (2.30), we see that \( G(L) \) and \( G(R) \) are skewness-adjusted quantiles of Student’s \( t \)-distribution for the left- and right-hand subintervals of N-Skart’s CI for \( \mu_X \). From (2.31) and (2.32), we see that the autoregression (correlation) adjustment \( A \) is applied to the sample \( S^2_{m,k'} \) defined by (2.27) so as to compensate for any residual correlation between the truncated, nonspaced batch means (2.26), that are used to compute the truncated grand mean \( \bar{Y}(m, k') \). A detailed algorithmic statement of N-Skart is given in Tafazzoli, Steiger, and Wilson (2010). Figure (2.3.2) depicts a high-level flowchart of the operation of N-Skart. In Chapter 3, we evaluate both the point and the CI estimator of \( \mu_X \) delivered by N-Skart for a wide range of test processes.

Figure 2.3: Flowchart of N-Skart
Chapter 3

Performance Comparison of N-Skart and MSER-5

3.1. Measures for Comparison of N-Skart and MSER-5

Several approaches have been proposed to evaluate the effects of initial conditions on the point estimator of a simulation output process. For our purpose, test problems have been chosen which are representative of the level of complexity observed in real-world systems. Some test problems have also been selected that exhibit extreme stochastic behavior and hence, can be used to stress test the two procedures. All these test cases have steady-state parameters (such as the steady-state mean) that can be obtained through analytical procedures; and therefore we can compare the performance of N-Skart and MSER5 with respect to their accuracy and precision in estimating these steady-state measures.

The effectiveness of the truncation methods in estimating the steady-state characteristics can be measured in terms of the following properties of confidence intervals computed with each procedure using nominal coverage probabilities (confidence coefficients) of 90% and 95%:

- Confidence-interval coverage measures the relative frequency with which the true mean of the process falls within the estimated confidence interval;

- Confidence-interval relative precision is the ratio of the CI half-length to the magnitude of the corresponding point estimator (usually the CI’s midpoint);

- Confidence-interval half-length measures the precision of the CI estimator; and

- Variance of the confidence-interval half-length measures the variability of the CI estimator.

The effectiveness of the two methods in removing the initial transient can also be measured in terms of the bias, the variance and mean-squared error of the truncated sample mean delivered by each
method. The bias measures systematic deviation of the point estimator away from the true mean, while the variance measures the random variation around the point estimator’s expected value. Truncation methods require considerably smaller sample sizes to reduce the bias in the simulation output to a specified level as compared to other methods for handling the simulation start-up problem. However, this may result in a significant increase in the variance (Fishman 1972). The combined effect on the point estimator due to the bias and the variance can be expressed in terms of a single numerical value as the mean-squared error, which can be expressed as

\[ \text{MSE}[\bar{Y}(m,k')] = \text{Bias}^2[\bar{Y}(m,k')] + \text{Var}[\bar{Y}(m,k')] \]

(3.1)

when we are evaluating the performance of the truncated sample mean \( \bar{Y}(m,k') \) delivered by N-Skart; and an equation similar to (3.1) applies to the truncated sample mean \( Z(k,d^*) \) delivered by MSER-5. We will use the bias, variance and mean-squared error as the measures for comparison of N-Skart and MSER-5.

For the purpose of our study, we will conduct 1,000 replications of each of the test cases. We will compare the average performance of N-Skart and MSER-5 over these replications in terms of their effectiveness in estimating a truncation point that minimizes the effect of initialization bias. We will also compare the performance of the two methods for simulation outputs of size \( N = 10,000, 20,000, 50,000 \) and 200,000, respectively, which would help us in investigating the relative performance of these methods for small, medium and large sample sizes. Most importantly, we will evaluate the performance of MSER-5 and N-Skart for the same set of output processes in order to block out the variability inherent in different realizations of each test process.

We shall consider the following statistics for comparison of the performance of N-Skart and MSER-5:

- Confidence-interval coverage;
- Average relative precision of the confidence interval;
- Average confidence-interval half-length;
- Variance of confidence-interval half-length;
- Truncated sample mean;
- Variance of the truncated sample mean; and
- Mean-squared error of the truncated sample mean.

In addition to above statistics, we will also consider the distribution of the truncated sample means obtained from 1,000 replications for each of the test cases. The distribution is approximated by the
histograms depicting the frequency distribution of the truncated sample means for sample sizes 10,000, 20,000, 50,000, and 200,000 respectively. Each histogram has been constructed from 1,000 i.i.d replications of the associated test process. The properties of the distribution depend upon the initial conditions and the sample size. Also, since both the methods deliver an estimate of the steady-state mean that is approximately free of any initialization bias, the resulting distribution of the truncated sample means should be representative of a limiting distribution that is attained as the number of replications goes to infinity. The test problems used for our analysis are described in the following section.

3.2. Description of Test Problems

3.2.1. $M/M/1$ Queue-Waiting-Time Process with Empty Initial Condition and 90% Utilization

The $M/M/1$ queue provides the first test process $\{X_j\}$ in which $X_j$ denotes the waiting time in the queue for the $j$th customer, where $j = 1, \ldots, N$ and $N = 10,000, 20,000, 50,000, 200,000$. The interarrival times for the customers are randomly sampled from an exponential distribution with an arrival rate of $\lambda = 0.9$ customers per unit time, and the service times for customers is sampled from an exponential distribution with a service rate of $\mu = 1.0$ customers per unit time. Thus, the steady-state server utilization for this process is given by

$$\rho = \frac{\lambda}{\mu} = \frac{0.9}{1.0} = 0.9.$$ (3.2)

The system starts in the empty-and-idle state so that $X_1 = 0$ on every replication of the process $\{X_j\}$. The steady-state expected waiting time in queue is given by

$$\mu_X = \lim_{j \to \infty} E(X_j) = \frac{\rho}{1 - \rho} = 9.0 \text{ time units.}$$ (3.3)

The $M/M/1$ queue process is characterized by a short warm-up period. However, the process exhibits a strong autocorrelation structure, with the autocorrelation function for the waiting time lags decaying slowly with increasing lags. Also, the $M/M/1$ queue waiting times have a steady-state probability distribution which has a nonzero probability at zero and an exponential tail. This results in a slow convergence of the batch means to the normal distribution with increasing batch size.

Figure 3.1 depicts a single realization of the first $M/M/1$ queue-waiting-time process. Table 3.1 summarizes the performance of the original MSER-5 algorithm, the modified MSER-5 algorithm, and N-Skart on 1,000 independent replications of this test process. Figures 3.2—3.5 display the empirical distribution of the truncated sample mean delivered by all three procedures in this test process for the selected sample sizes.
Figure 3.1: A realization of the $M/M/1$ queue-waiting-time process with empty-and-idle initial condition and 90% server utilization
Table 3.1: Performance of MSER-5 and N-Skart in the $M/M/1$ queue-waiting-time process with 90% server utilization and empty-and-idle initial condition

<table>
<thead>
<tr>
<th></th>
<th>Overall Sample Size N</th>
<th>N = 10,000</th>
<th>N = 20,000</th>
<th>N = 50,000</th>
<th>N = 200,000</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confidence-Interval Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1 - \alpha$</td>
<td>Empirical Perf. Meas.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90% CI coverage</td>
<td>8.00%</td>
<td>11.50%</td>
<td>12.80%</td>
<td>14.30%</td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>6.80%</td>
<td>5.04%</td>
<td>3.60%</td>
<td>1.23%</td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.28063</td>
<td>0.21491</td>
<td>0.14611</td>
<td>0.078336</td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.0092481</td>
<td>0.0043447</td>
<td>0.0012732</td>
<td>0.00015273</td>
<td></td>
</tr>
<tr>
<td>95% CI coverage</td>
<td>10.00%</td>
<td>13.20%</td>
<td>15.20%</td>
<td>17.40%</td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>8.10%</td>
<td>6.01%</td>
<td>4.29%</td>
<td>1.47%</td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.33437</td>
<td>0.25667</td>
<td>0.17409</td>
<td>0.093337</td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.013129</td>
<td>0.0061680</td>
<td>0.0018075</td>
<td>0.00021682</td>
<td></td>
</tr>
<tr>
<td><strong>Empirical Point-Estimator</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall Sample Size N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
<td>7.6637</td>
<td>8.0418</td>
<td>8.2466</td>
<td>8.7850</td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>24.286</td>
<td>17.030</td>
<td>11.778</td>
<td>4.8624</td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td>22.501</td>
<td>16.112</td>
<td>11.210</td>
<td>4.8161</td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>1.3363</td>
<td>0.95818</td>
<td>0.75342</td>
<td>0.21498</td>
<td></td>
</tr>
</tbody>
</table>

Results for Modified MSER-5 Algorithm

<table>
<thead>
<tr>
<th></th>
<th>Overall Sample Size N</th>
<th>N = 10,000</th>
<th>N = 20,000</th>
<th>N = 50,000</th>
<th>N = 200,000</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confidence-Interval Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1 - \alpha$</td>
<td>Empirical Perf. Meas.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90% CI coverage</td>
<td>10.60%</td>
<td>13.50%</td>
<td>14.60%</td>
<td>15.40%</td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>5.15%</td>
<td>3.35%</td>
<td>2.07%</td>
<td>0.90%</td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.31470</td>
<td>0.23783</td>
<td>0.15603</td>
<td>0.080708</td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.0045737</td>
<td>0.0016004</td>
<td>0.00037845</td>
<td>3.7545E-05</td>
<td></td>
</tr>
<tr>
<td>95% CI coverage</td>
<td>13.10%</td>
<td>15.70%</td>
<td>17.10%</td>
<td>18.70%</td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>6.14%</td>
<td>3.99%</td>
<td>2.47%</td>
<td>1.07%</td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.37496</td>
<td>0.28337</td>
<td>0.18590</td>
<td>0.096163</td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.0064930</td>
<td>0.0022720</td>
<td>0.00053726</td>
<td>5.3300E-05</td>
<td></td>
</tr>
<tr>
<td><strong>Empirical Point-Estimator</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall Sample Size N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
<td>7.7038</td>
<td>8.3265</td>
<td>8.6895</td>
<td>8.9591</td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>24.286</td>
<td>17.030</td>
<td>11.778</td>
<td>4.8624</td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td>22.501</td>
<td>16.112</td>
<td>11.210</td>
<td>4.8161</td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>1.3363</td>
<td>0.95818</td>
<td>0.75342</td>
<td>0.21498</td>
<td></td>
</tr>
</tbody>
</table>

Results for N-Skart Algorithm

<table>
<thead>
<tr>
<th></th>
<th>Overall Sample Size N</th>
<th>N = 10,000</th>
<th>N = 20,000</th>
<th>N = 50,000</th>
<th>N = 200,000</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Confidence-Interval Properties</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1 - \alpha$</td>
<td>Empirical Perf. Meas.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90% CI coverage</td>
<td>87.70%</td>
<td>91.80%</td>
<td>92.50%</td>
<td>92.10%</td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>43.07%</td>
<td>32.50%</td>
<td>20.71%</td>
<td>9.51%</td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>4.0592</td>
<td>3.0594</td>
<td>1.8932</td>
<td>0.86168</td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>10.350</td>
<td>9.2895</td>
<td>1.4728</td>
<td>0.07406</td>
<td></td>
</tr>
<tr>
<td>95% CI coverage</td>
<td>92.90%</td>
<td>95.70%</td>
<td>95.50%</td>
<td>96.30%</td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>57.05%</td>
<td>42.81%</td>
<td>26.93%</td>
<td>11.75%</td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>5.3664</td>
<td>4.0040</td>
<td>2.4653</td>
<td>1.0655</td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>19.244</td>
<td>17.733</td>
<td>3.1065</td>
<td>0.18158</td>
<td></td>
</tr>
<tr>
<td><strong>Empirical Point-Estimator</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall Sample Size N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
<td>8.9055</td>
<td>8.9445</td>
<td>8.9625</td>
<td>9.0062</td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>3.5225</td>
<td>1.6052</td>
<td>0.65584</td>
<td>0.17491</td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td>3.5136</td>
<td>1.6022</td>
<td>0.65444</td>
<td>0.17487</td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>0.09455</td>
<td>0.055517</td>
<td>0.037485</td>
<td>0.0061757</td>
<td></td>
</tr>
</tbody>
</table>
Figure 3.2: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0$, $\rho = 0.9$, and $N = 10,000$
Figure 3.3: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0$, $\rho = 0.9$, and $N = 20,000$
Figure 3.4: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0$, $\rho = 0.9$, and $N = 50,000$. 
Figure 3.5: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0$, $\rho = 0.9$, and $N = 200,000$.
Consider the results for $M/M/1$ queue with 90% server utilization and empty and idle initial condition. The steady-state mean of this process is 9.0. From the results in Table 3.1, it is evident that the confidence-interval properties obtained from N-Skart were considerably better than those obtained from both the MSER-5 algorithms. The confidence interval coverages delivered by N-Skart were extremely close to the specified coverages. For smaller sample sizes, N-Skart delivered results with lesser CI coverages than specified; but it also delivered a CI with wider relative precision, indicating that a larger sample size was required in order to have practically useful CIs. As the sample size increased, N-Skart delivered CI estimates close to the specified coverage and in most cases, the actual coverage was better than the specified CI coverage as is observed in the case of sample size 200,000 and specified coverage of 95%. As against this, the CI delivered by both the MSER-5 algorithms exhibited very low CI coverages, typically in the range 10% – 20%. The average relative precision and the actual CI half-length delivered by N-Skart was almost an order of magnitude higher than that delivered by MSER-5, which indicates that in addition to providing a valid CI for the point estimator, N-Skart also provides a much more realistic estimate of the true precision of the truncated sample mean (as measured by the CI relative precision or the CI average half-length) than either version of MSER-5 provides.

While considering the point estimator delivered by both the methods, it was observed that the MSER-5 algorithms greatly underestimated the value of the steady-state mean. The MSE and variance in the estimate of the steady-state mean was at least an order of magnitude smaller for N-Skart than MSER-5, for all sample sizes. The modified MSER-5 provided improved results over the original MSER-5 by reducing the MSE by an order of magnitude for smaller sample sizes, but these values were still substantially larger than those delivered by N-Skart. Another point worth mentioning here is that for both N-Skart and MSER-5, the variance was the main contributor in the MSE.

The bias and variance in the estimate of the steady-state mean can also be observed in the histograms depicted in Figures 3.2—3.5. The truncated sample means were distributed over a much wider range in the case of both the MSER-5 algorithms. Also, for smaller sample sizes, the distribution exhibited bimodal characteristics, and the mode was shifted to the left of the distribution. It was also observed that MSER-5 delivered mean estimates as high as 46.44, which is completely unrepresentative of the steady-state mean of this test process. Moreover, for sample size 10,000, the upper and lower quartiles of the distributions did not contain the steady-state value (i.e. 9) in the case of the MSER-5 algorithms. All in all, N-Skart greatly outperformed both versions of MSER-5 in this test process.
3.2.2. $M/M/1$ Queue-Waiting-Time Process with Empty Initial Condition and 80% Utilization

This test process is similar to the first test process but the interarrival times are sampled from an exponential distribution with an arrival rate of $\lambda = 0.8$ arrivals per unit time. The steady-state server utilization for this process is given by

$$\rho = \frac{\lambda}{\mu} = \frac{0.8}{1.0} = 0.8. \quad (3.4)$$

The system starts in the empty-and-idle state so that $X_1 = 0$ on every replication of the process $\{X_j\}$. Thus, the steady-state expected waiting time for this process is given by

$$\mu_X = \lim_{j \to \infty} E(X_j) \frac{\rho}{1 - \rho} = 4.0 \text{ time units} \quad (3.5)$$

Figure 3.6 depicts a single realization of the $M/M/1$ queue-waiting-time process with empty-and-idle initial condition and 80% server utilization. Table 3.2 summarizes the performance of MSER-5, modified MSER-5, and N-Skart on 1,000 independent replications of this test process. Figures 3.7—3.10 display the empirical distributions of the truncated sample mean delivered by all three procedures in this test process. The results for this process are given below.

Figure 3.6: A realization of the $M/M/1$ queue-waiting-time process with empty-and-idle initial condition and 80% server utilization
Table 3.2: Performance of MSER-5 and N-Skart in the $M/M/1$ queue-waiting-time process with 80% server utilization and empty-and-idle initial condition

<table>
<thead>
<tr>
<th>Results for Original MSER-5 Algorithm</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidence-Interval Properties</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1 - \alpha$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI coverage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI Coverage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empirical Point-Estimator</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Measures</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI Coverage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empirical Point-Estimator</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Measures</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Results for Modified MSER-5 Algorithm</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidence-Interval Properties</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1 - \alpha$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI coverage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI Coverage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empirical Point-Estimator</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Measures</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI Coverage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empirical Point-Estimator</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Measures</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Results for N-Skart Algorithm</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidence-Interval Properties</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$1 - \alpha$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI coverage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CI Coverage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Empirical Point-Estimator</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Measures</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

31
Figure 3.7: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0$, $\rho = 0.8$, and $N = 10,000$. 
Figure 3.8: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0$, $\rho = 0.8$, and $N = 20,000$
Figure 3.9: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0$, $\rho = 0.8$, and $N = 50,000$. 
Figure 3.10: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with $X_1 = 0$, $\rho = 0.8$, and $N = 200,000$.
The $M/M/1$ queue-waiting-time process with empty-and-idle initial condition and 80% server utilization has a much smaller initial transient than the first test process. The steady-state mean waiting time in queue for this process is 4. From the results obtained in Table 3.2, we can see that the CI coverages delivered by both the MSER-5 algorithms were better than those delivered for the first test process, but the coverages were still lower (20% – 40%) than the specified CI coverage levels. The other confidence-interval properties followed the same trends as the previous test problem.

The estimates provided by N-Skart for the MSE, variance and bias of the truncated sample mean were considerably better than those provided by both the MSER-5 algorithms. In particular, the values for bias delivered by N-Skart were at least an order of magnitude smaller for all sample sizes. With increasing sample size, both the methods delivered an estimate of the steady-state mean with an improved value of the MSE.

The truncated sample means were more tightly distributed about the steady-state mean as compared to the previous test process. With increasing sample size, the variability in the truncated sample means delivered by N-Skart and both the MSER-5 algorithms reduced considerably. All in all, we concluded that N-Skart significantly outperformed both versions of MSER-5 in this test process. For moderate ($N = 50,000$) and large ($N = 200,000$) sample sizes, the modified MSER-5 delivered a point estimator of $\mu_X$ with performance nearly as good as that of N-Skart; but the CIs delivered by the modified MSER-5 had unacceptable coverage probabilities while the CIs delivered by N-Skart had coverage probabilities close to the nominal levels.
3.2.3. $M/M/1$ Queue-Waiting-Time Process with 113 Initial Customers and 90% Utilization

This test process has the same interarrival rate and service rate as the first test process (i.e. $\lambda = 0.9$ and $\mu = 1$), but with an additional condition that 113 customers are already present in queue at time 0. The first “regular” customer arrives after time 0 and begins service after the initial 113 customers have finished service. This initial condition ensures that the expected queue waiting time for the first regular customer to arrive after time 0 is 10 steady-state standard deviations above the steady-state mean, ensuring a pronounced initial transient for this test process as explained in Tafazzoli (2009). This process has the same steady-state parameters as the first test process—i.e., a server utilization of $\rho = 0.9$ and steady-state mean queue-waiting time of $\mu_X = 9.0$ time units. The results for this process are given below.

Figure 3.11 depicts a single realization of the $M/M/1$ queue-waiting-time process with 113 initial customers and 90% server utilization. Figure 3.12 depicts the transient behavior of this process for three independent replications. Table 3.3 summarizes the performance of MSER-5, modified MSER-5, and N-Skart on 1,000 independent replications of this test process. Figures 3.13—3.16 display the empirical distributions of the truncated sample mean delivered by all three procedures in this test process.

![Figure 3.11: A realization of the $M/M/1$ queue-waiting time process with 113 initial customers and 90% server utilization](image-url)
Figure 3.12: A depiction of the transient behavior of the $M/M/1$ queue-waiting time process with 113 initial customers and 90% server utilization for 3 independent replications.
Table 3.3: Performance of MSER-5 and N-Skart in the M/M/1 queue-waiting-time process with 90% server utilization and 113 initial customers

<table>
<thead>
<tr>
<th>Results for Original MSER-5 Algorithm</th>
<th>Overall Sample Size N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 − α Empirical Perf. Meas.</td>
<td>10,000 20,000 50,000 200,000</td>
</tr>
<tr>
<td>CI coverage</td>
<td>6.00% 11.60% 13.50% 14.00%</td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>9.09% 5.74% 3.51% 1.54%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.28332 0.21975 0.14768 0.077074</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.011190 0.0044192 0.0013191 0.00022193</td>
</tr>
<tr>
<td>CI Coverage</td>
<td>7.80% 14.20% 14.90% 16.90%</td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>10.83% 6.84% 4.18% 1.84%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.33757 0.26183 0.17596 0.091833</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.015886 0.0062737 0.0018726 0.00031506</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Results for Modified MSER-5 Algorithm</th>
<th>Overall Sample Size N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 − α Empirical Perf. Meas.</td>
<td>10,000 20,000 50,000 200,000</td>
</tr>
<tr>
<td>CI coverage</td>
<td>8.10% 13.80% 15.60% 15.40%</td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>5.90% 3.52% 2.06% 0.90%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.32579 0.24255 0.15760 0.080887</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.0056589 0.0016885 0.00038284 3.7835E-05</td>
</tr>
<tr>
<td>CI Coverage</td>
<td>10.40% 16.60% 17.30% 19.10%</td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>7.03% 4.20% 2.45% 1.07%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>0.38817 0.28900 0.18778 0.096376</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.0080336 0.0023971 0.00054350 5.3712E-05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Results for N-Skart Algorithm</th>
<th>Overall Sample Size N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 − α Empirical Perf. Meas.</td>
<td>10,000 20,000 50,000 200,000</td>
</tr>
<tr>
<td>CI coverage</td>
<td>96.00% 95.10% 92.60% 91.90%</td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>126.96% 69.83% 27.81% 10.06%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>16.186 7.5610 2.6154 0.91283</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>188.26 50.895 7.043 0.27369</td>
</tr>
<tr>
<td>CI Coverage</td>
<td>98.60% 98.80% 97.00% 97.10%</td>
</tr>
<tr>
<td>Avg. rel. prec.</td>
<td>182.48% 97.79% 37.44% 12.56%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>23.549 10.614 3.5296 1.1402</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>419.36 95.930 12.160 0.41319</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Empirical Point-Estimator</th>
<th>Overall Sample Size N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trunc. Sample Mean</td>
<td>7.1428 7.9350 8.4248 8.5610</td>
</tr>
<tr>
<td>MSE</td>
<td>27.142 18.602 14.0374 5.4136</td>
</tr>
<tr>
<td>Variance</td>
<td>23.693 17.468 13.7065 5.2209</td>
</tr>
<tr>
<td>Bias</td>
<td>1.8572 1.0650 0.57524 0.43897</td>
</tr>
<tr>
<td>MSE</td>
<td>7.4551 8.3133 8.7089 8.9594</td>
</tr>
<tr>
<td>Variance</td>
<td>6.2836 2.8311 1.1488 0.17642</td>
</tr>
<tr>
<td>Bias</td>
<td>1.5449 0.68668 0.29113 0.404612</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Empirical Point-Estimator</th>
<th>Overall Sample Size N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trunc. Sample Mean</td>
<td>11.985 10.201 9.1133 9.0141</td>
</tr>
<tr>
<td>MSE</td>
<td>22.551 5.2009 0.76609 0.17666</td>
</tr>
<tr>
<td>Variance</td>
<td>13.643 3.7579 0.75325 0.17646</td>
</tr>
<tr>
<td>Bias</td>
<td>2.9846 1.2012 0.11334 0.014065</td>
</tr>
</tbody>
</table>
Figure 3.13: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with 113 initial customers, $\rho = 0.9$, and $N = 10,000$
Figure 3.14: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with 113 initial customers, $\rho = 0.9$, and $N = 20,000$
Figure 3.15: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with 113 initial customers, $\rho = 0.9$, and $N = 50,000$. 
Figure 3.16: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to $M/M/1$ queue-waiting-time process with 113 initial customers, $\rho = 0.9$, and $N = 200,000$
This process is characterized by a large initial transient period and serves as an extreme test problem for all three methods. Now, consider the results for the $M/M/1$ queue with 113 initial customers given in Table 3.3. It was observed that the CI coverages delivered by both the MSER-5 versions were considerably lower than for the previous test problems. As against this, the CI coverages provided by N-Skart were considerably better than the nominal coverage probabilities. However, for smaller sample sizes, N-Skart delivered extremely large CI half-widths, indicating that larger sample sizes were required to obtain practically useful results. With increasing sample size, the average CI half-width delivered by N-Skart reduced by an order of magnitude for successive increases in the sample size, while still conforming to the specified CI precision levels.

Based on all our computational experience with N-Skart, we have found that the point and CI estimators of $\mu_X$ are generally reliable provided that the relative precision of the delivered CI of the form $\bar{Y}(m,k') \pm H$ satisfies

$$H/|\bar{Y}(m,k')| \leq 0.40; \quad (3.6)$$

and when (3.6) is not satisfied, this is a strong indication that the user should increase the size $N$ of the data set substantially—that is, until (3.6) is finally satisfied.

It was also observed that for sample sizes of 10,000 and 20,000, the modified MSER-5 algorithm produced results with MSE and variance that were an order of magnitude smaller than those delivered by N-Skart or the original MSER-5. It is noteworthy that the condition (3.6) for reliable use of N-Skart’s results is not satisfied for the sample sizes $N = 10,000$ and $N = 20,000$. On the other hand, condition (3.6) is satisfied for the sample sizes $N = 50,000$ and $N = 200,000$; and in these cases, N-Skart significantly outperformed MSER-5 and modified MSER-5 with respect to point-estimator accuracy. However, for larger sample sizes, N-Skart outperformed both the MSER-5 versions. Another point worth mentioning here is that for $M/M/1$ queue with 113 initial customers, N-Skart overestimated the steady-state mean while both the MSER-5 algorithms underestimated the steady-state mean.

The distribution of the truncated sample means for N-Skart was considerably skewed as compared to that of the modified MSER-5. The original MSER-5 exhibited higher variability and bimodal characteristics and the variability was not reduced substantially with increasing sample sizes. Also, none of the methods contained the true steady-state mean between the upper and lower quartiles for the case of $N = 10,000$. With increasing sample size, the distribution of the truncated sample mean obtained from N-Skart was closely distributed about the true mean of the test process. Similar improvement was also observed in the modified MSER-5 results.

Overall, we found that N-Skart outperformed MSER-5 in this test process. For small sample sizes ($N = 10,000$ and $N = 20,000$), the point estimator delivered by modified MSER-5 outperformed that of N-Skart; but for medium and large sample sizes ($N = 50,000$ and $N = 200,000$), the position was reversed. The CIs delivered by both versions of MSER-5 had unacceptable coverage probabilities, while
the CIs delivered by N-Skart had coverage probabilities that slightly exceeded their nominal levels.

### 3.2.4. First-Order Autoregressive (AR(1)) Process

The next test process is the first-order autoregressive (AR(1)) process,

\[
X_j = \mu_X + \rho(X_{j-1} - \mu_X) + \varepsilon_j \quad \text{for} \quad j = 1, \ldots, N,
\]

where initial condition \(X_0 = 0\) and the autoregressive parameter \(\rho = 0.995\), and the steady-state mean \(\mu_X = 100\). The error terms \(\{\varepsilon_j : j = 1, 2, 3, \ldots\}\) are randomly sampled from a \(N(0, 1)\) distribution. The results for this process are given below. Figure 3.17 depicts a realization of the AR(1) process (3.7) with \(X_0 = 0\), \(\mu_X = 100\), and \(\rho = 0.995\). This initial condition ensures that the expected value of the first observation after time 0 is approximately 10 steady-state standard deviations below the steady-state mean, ensuring a pronounced initial transient for this test process as explained in Tafazzoli (2009). Figure 3.18 depicts the transient behavior of this process for three independent replications. Table 3.4 summarizes the performance of MSER-5, modified MSER-5, and N-Skart on 1,000 independent replications of this test process. Figures 3.19—3.22 display the empirical distributions of the truncated sample means delivered by all three procedures in this test process.

![Figure 3.17: A realization of the AR(1) Process (3.7) \(X_0 = 0\), \(\mu_X = 100\), and \(\rho = 0.995\)](image-url)
Figure 3.18: A depiction of the transient behavior of the AR(1) Process (3.7) $X_0 = 0$, $\mu_X = 100$, and $\rho = 0.995$ for 3 independent replications
Table 3.4: Performance of MSER-5 and N-Skart in the AR(1) process

<table>
<thead>
<tr>
<th>Confidence-Interval Properties</th>
<th>Overall Sample Size</th>
<th>10,000</th>
<th>20,000</th>
<th>50,000</th>
<th>200,000</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Results for Original MSER-5 Algorithm</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Confidence-Interval Properties</strong></td>
<td>Empirical Perf. Meas.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 – α</td>
<td>CI coverage</td>
<td>10.20%</td>
<td>11.40%</td>
<td>12.60%</td>
<td>12.70%</td>
</tr>
<tr>
<td>90%</td>
<td>Avg. rel. prec.</td>
<td>0.31%</td>
<td>0.23%</td>
<td>0.15%</td>
<td>0.08%</td>
</tr>
<tr>
<td></td>
<td>Avg. CI half-length</td>
<td>0.31062</td>
<td>0.23295</td>
<td>0.15150</td>
<td>0.079932</td>
</tr>
<tr>
<td></td>
<td>Var. CI half-length</td>
<td>0.010141</td>
<td>0.0036738</td>
<td>0.0012981</td>
<td>0.00011158</td>
</tr>
<tr>
<td>95%</td>
<td>CI Coverage</td>
<td>12.50%</td>
<td>14.30%</td>
<td>15.20%</td>
<td>14.80%</td>
</tr>
<tr>
<td></td>
<td>Avg. rel. prec.</td>
<td>0.37%</td>
<td>0.28%</td>
<td>0.18%</td>
<td>0.10%</td>
</tr>
<tr>
<td></td>
<td>Avg. CI half-length</td>
<td>0.37010</td>
<td>0.27756</td>
<td>0.18050</td>
<td>0.095238</td>
</tr>
<tr>
<td></td>
<td>Var. CI half-length</td>
<td>0.014937</td>
<td>0.0052154</td>
<td>0.0018429</td>
<td>0.00015840</td>
</tr>
<tr>
<td><strong>Empirical Point-Estimator</strong></td>
<td>Performance Measures</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall Sample Size N</td>
<td>10,000</td>
<td>20,000</td>
<td>50,000</td>
<td>200,000</td>
<td></td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
<td>99.814</td>
<td>99.787</td>
<td>100.05</td>
<td>100.05</td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>26.704</td>
<td>21.090</td>
<td>17.570</td>
<td>6.1289</td>
<td></td>
</tr>
<tr>
<td>Bias</td>
<td>0.18598</td>
<td>0.21288</td>
<td>0.15150</td>
<td>0.079932</td>
<td></td>
</tr>
</tbody>
</table>

| **Results for Modified MSER-5 Algorithm** |                      |        |        |        |         |
| **Confidence-Interval Properties** | Empirical Perf. Meas. |        |        |        |         |
| 1 – α | CI coverage | 13.50% | 14.10% | 14.90% | 13.60%  |
| 90%   | Avg. rel. prec. | 0.36%  | 0.26%  | 0.16%  | 0.08%   |
|       | Avg. CI half-length | 0.35993 | 0.25698 | 0.16385 | 0.08203 |
|       | Var. CI half-length | 0.001502 | 0.00036783 | 5.5514E-05 | 3.2774E-06 |
| 95%   | CI Coverage | 16.70% | 17.30% | 17.80% | 15.70%  |
|       | Avg. rel. prec. | 0.43%  | 0.31%  | 0.20%  | 0.10%   |
|       | Avg. CI half-length | 0.42886 | 0.30619 | 0.19523 | 0.097738 |
|       | Var. CI half-length | 0.0021323 | 0.00052218 | 7.8810E-05 | 4.6528E-06 |
| **Empirical Point-Estimator** | Performance Measures |        |        |        |         |
| Overall Sample Size N | 10,000 | 20,000 | 50,000 | 200,000 |
| MSE | 7.0888 | 2.4980 | 0.77769 | 0.20182 |
| Variance | 7.0662 | 2.4802 | 0.77662 | 0.20171 |
| Bias | 0.15025 | 0.13311 | 0.03273 | 0.010679 |

| **Results for N-Skart Algorithm** |                      |        |        |        |         |
| **Confidence-Interval Properties** | Empirical Perf. Meas. |        |        |        |         |
| 1 – α | CI coverage | 91.30% | 93.50% | 94.20% | 93.10%  |
| 90%   | Avg. rel. prec. | 4.31%  | 2.86%  | 1.74%  | 0.82%   |
|       | Avg. CI half-length | 4.2818 | 2.8555 | 1.7405 | 0.8173 |
|       | Var. CI half-length | 0.65708 | 0.19415 | 0.033968 | 0.0022974 |
| 95%   | CI Coverage | 95.30% | 97.50% | 98.00% | 96.50%  |
|       | Avg. rel. prec. | 5.26%  | 3.46%  | 2.09%  | 0.98%   |
|       | Avg. CI half-length | 5.2275 | 3.4461 | 2.0897 | 0.97699 |
|       | Var. CI half-length | 0.99768 | 0.30315 | 0.049615 | 0.0022974 |
| **Empirical Point-Estimator** | Performance Measures |        |        |        |         |
| Overall Sample Size N | 10,000 | 20,000 | 50,000 | 200,000 |
| MSE | 4.3722 | 2.1129 | 0.75410 | 0.20109 |
| Variance | 4.0845 | 2.0426 | 0.75329 | 0.20105 |
| Bias | 0.53636 | 0.26523 | 0.028682 | 0.0064897 |
Figure 3.19: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to AR(1) process (3.7) with $X_0 = 0$, $\mu_X = 100$, $\rho = 0.995$, and $N = 10,000$
Figure 3.20: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to AR(1) process (3.7) with $X_0 = 0$, $\mu_X = 100$, $\rho = 0.995$, and $N = 20,000$.
Figure 3.21: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to AR(1) process (3.7) with $X_0 = 0$, $\mu_X = 100$, $\rho = 0.995$, and $N = 50,000$. 

<table>
<thead>
<tr>
<th>Quantiles</th>
<th>N-Skart</th>
<th>MSER5 w/o correction</th>
<th>MSER5 with correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.0% maximum</td>
<td>102.578</td>
<td>126.711</td>
<td>102.777</td>
</tr>
<tr>
<td>99.5%</td>
<td>102.18</td>
<td>122.728</td>
<td>102.329</td>
</tr>
<tr>
<td>97.5%</td>
<td>101.7</td>
<td>110.775</td>
<td>101.714</td>
</tr>
<tr>
<td>90.0%</td>
<td>101.066</td>
<td>101.464</td>
<td>101.101</td>
</tr>
<tr>
<td>75.0% quartile</td>
<td>99.992</td>
<td>100.636</td>
<td>99.4748</td>
</tr>
<tr>
<td>50.0% median</td>
<td>99.962</td>
<td>99.2348</td>
<td>99.3424</td>
</tr>
<tr>
<td>25.0% quartile</td>
<td>99.349</td>
<td>98.4959</td>
<td>98.8438</td>
</tr>
<tr>
<td>10.0%</td>
<td>98.8637</td>
<td>89.9615</td>
<td>98.2431</td>
</tr>
<tr>
<td>2.5%</td>
<td>98.2984</td>
<td>80.6646</td>
<td>0.5%</td>
</tr>
<tr>
<td>0.5% minimum</td>
<td>97.6495</td>
<td>75.4762</td>
<td>97.6307</td>
</tr>
<tr>
<td>0.0% minimum</td>
<td>96.986</td>
<td>75.4762</td>
<td>96.9738</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Moments</th>
<th>N-Skart</th>
<th>MSER5 w/o correction</th>
<th>MSER5 with correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>99.971318</td>
<td>100.05157</td>
<td>99.967275</td>
</tr>
<tr>
<td>Std Dev</td>
<td>0.8653543</td>
<td>4.1933886</td>
<td>0.8817022</td>
</tr>
<tr>
<td>Std Err Mean</td>
<td>0.0274598</td>
<td>0.1326066</td>
<td>0.0278819</td>
</tr>
<tr>
<td>Upper 95% Mean</td>
<td>100.0252</td>
<td>100.31179</td>
<td>100.02199</td>
</tr>
<tr>
<td>Lower 95% Mean</td>
<td>99.917433</td>
<td>99.791355</td>
<td>99.912561</td>
</tr>
<tr>
<td>N</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>
Figure 3.22: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to AR(1) process (3.7) with $X_0 = 0$, $\mu_X = 100$, $\rho = 0.995$, and $N = 200,000$. 

Distributions

<table>
<thead>
<tr>
<th>N-Skart</th>
<th>MSER5 w/o correction</th>
<th>MSER5 with correction</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Quantiles</strong></td>
<td><strong>Quantiles</strong></td>
<td><strong>Quantiles</strong></td>
</tr>
<tr>
<td>100.0% maximum</td>
<td>101.635</td>
<td>101.623</td>
</tr>
<tr>
<td>99.5%</td>
<td>101.207</td>
<td>101.217</td>
</tr>
<tr>
<td>97.5%</td>
<td>100.886</td>
<td>100.873</td>
</tr>
<tr>
<td>90.0%</td>
<td>100.544</td>
<td>100.545</td>
</tr>
<tr>
<td>75.0% quartile</td>
<td>100.296</td>
<td>100.296</td>
</tr>
<tr>
<td>50.0% median</td>
<td>100.012</td>
<td>100.008</td>
</tr>
<tr>
<td>25.0% quartile</td>
<td>99.6723</td>
<td>99.669</td>
</tr>
<tr>
<td>10.0%</td>
<td>99.4124</td>
<td>99.4063</td>
</tr>
<tr>
<td>2.5%</td>
<td>98.8418</td>
<td>99.1233</td>
</tr>
<tr>
<td>0.5%</td>
<td>98.9148</td>
<td>0.5%</td>
</tr>
<tr>
<td>0.0% minimum</td>
<td>98.7761</td>
<td>0.0% minimum</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Moments</th>
<th>Moments</th>
<th>Moments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>99.99351</td>
<td>100.05165</td>
</tr>
<tr>
<td>Std Dev</td>
<td>0.4486093</td>
<td>2.4763541</td>
</tr>
<tr>
<td>Std Err Mean</td>
<td>0.0141863</td>
<td>0.0783092</td>
</tr>
<tr>
<td>Upper 95% Mean</td>
<td>100.02135</td>
<td>100.20532</td>
</tr>
<tr>
<td>Lower 95% Mean</td>
<td>99.965672</td>
<td>99.897983</td>
</tr>
<tr>
<td>N</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>
This process is characterized by a high level of positive correlation between successive observations and a large negative initialization bias which severely distorts the behavior of the classical batch means method. From the results, it is clear that N-Skart outperformed MSER-5 in terms of CI coverage for all the sample sizes. Although the average CI half-lengths delivered by N-Skart were an order of magnitude higher than those delivered by both the MSER-5 algorithms, the CI coverages delivered by N-Skart were better than the user-specified CI coverages.

The modified MSER-5 delivered values of MSE and variance that were comparable to those delivered by N-Skart for almost all the sample sizes. However, N-Skart delivered values of bias that were at least an order of magnitude smaller than those delivered by both the MSER-5 algorithms.

Histograms for the methods indicated that the original MSER-5 algorithm fared poorly in comparison with the modified MSER-5 algorithm and N-Skart. The distributions for all methods were symmetrically distributed about the mode and the distributions became more tightly distributed as the sample size increased. The variance for both the MSER-5 algorithms was considerably higher, especially for smaller sample sizes.
3.2.5. AR(1)-to-Pareto (ARTOP) Process

The AR(1)-to-Pareto Process is generated from an underlying AR(1) process,

\[ Z_j = \rho Z_{j-1} + b_j \quad \text{for} \quad j = 1, \ldots, N, \quad (3.8) \]

where \( \{b_j : j = 1, \ldots, N\} \) \( \overset{\text{i.i.d.}}{\sim} N(0, \sigma_b^2) \) is a white noise process with variance

\[ \sigma_b^2 = \sigma_Z^2(1 - \rho^2) = 1 - \rho^2. \quad (3.9) \]

This process is used to generate a set of correlated random variables \( \{U_j = \Phi(Z_i) : j = 1, \ldots, N\} \) whose marginal distribution is \( U \sim (0,1) \). where

\[ \Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{\zeta^2}{2}} d\zeta \quad \text{for all real} \quad z \quad (3.10) \]

is the \( N(0,1) \) c.d.f. The correlated sequence of random numbers \( \{U_j : j = 1, \ldots, N\} \) is supplied as input to the inverse of the Pareto c.d.f.

\[ F_X(x) \equiv \Pr\{X \leq x\} = \begin{cases} 1 - (\xi / x)^\psi, & x \geq \xi, \\ 0, & x < \xi, \end{cases} \quad (3.11) \]

where \( \xi > 0 \) is a location parameter and \( \psi > 0 \) is a shape parameter, to generate the ARTOP process \( \{X_j : j = 1, \ldots, N\} \) as follows

\[ X_j = F_X^{-1}(U_j) = F_X^{-1}[\Phi(Z_j)] = \frac{\xi}{[1 - \Phi(Z_j)]^{1/\psi}} \quad \text{for} \quad j = 1, \ldots, N \quad (3.12) \]

The mean and variance of the ARTOP process are given by

\[ \mu_X = \mathbb{E}[X_j] = \psi \xi (\psi - 1)^{-1} \quad (\text{for} \quad \psi < 1), \quad (3.13) \]

\[ \sigma_X^2 = \text{Var}[X_j] = \xi^2 \psi (\psi - 1)^{-2} (\psi - 2)^{-1} \quad (\text{for} \quad \psi > 2). \quad (3.14) \]

We choose \( \psi = 2.1 \) and \( \xi = 1.0 \) and the lag one correlation in the base process is set to \( \rho = 0.995 \), so that we obtain an ARTOP process whose mean \( \mu_X = 1.9091 \) and variance \( \sigma_X^2 = 17.3554 \). Also, we choose the initial condition \( Z_0 = 3.4 \) so that \( X_0 = F_X^{-1}[\Phi(Z_0)] = 43.569 \); and thus, we get a highly correlated nonnormal process with a large transient period.

Figure 3.23 depicts a realization of the ARTOP process (3.12). Figure 3.24 depicts the transient behavior of this process for three independent replications. Table 3.5 summarizes the performance
of MSER-5, modified MSER-5, and N-Skart on 1,000 independent replications of this test process. Figures 3.25—3.28 display the empirical distributions of the truncated sample means delivered by all these procedures in this test process.

Figure 3.23: A realization of the ARTOP Process
Figure 3.24: A depiction of the transient behavior of the ARTOP Process for 3 independent replications
Table 3.5: Performance of MSER-5 and N-Skart in the ARTOP process

### Results for Original MSER-5 Algorithm

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 − α</td>
<td>10,000</td>
<td>20,000</td>
</tr>
<tr>
<td>CI coverage</td>
<td>2.00%</td>
<td>3.00%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>1.43%</td>
<td>1.25%</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.021581</td>
<td>0.018736</td>
</tr>
<tr>
<td>CI coverage</td>
<td>2.30%</td>
<td>3.30%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>1.71%</td>
<td>1.49%</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.000044641</td>
<td>0.000031184</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Empirical Point-Estimator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Measures</td>
</tr>
<tr>
<td>Overall Sample Size N</td>
</tr>
<tr>
<td>10,000</td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
</tr>
<tr>
<td>MSE</td>
</tr>
<tr>
<td>Variance</td>
</tr>
<tr>
<td>Bias</td>
</tr>
</tbody>
</table>

### Results for Modified MSER-5 Algorithm

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 − α</td>
<td>10,000</td>
<td>20,000</td>
</tr>
<tr>
<td>CI coverage</td>
<td>2.70%</td>
<td>3.90%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>1.78%</td>
<td>1.56%</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.000032653</td>
<td>0.000020376</td>
</tr>
<tr>
<td>CI coverage</td>
<td>3.30%</td>
<td>4.40%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>2.12%</td>
<td>1.86%</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>0.000046356</td>
<td>0.000028927</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Empirical Point-Estimator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Measures</td>
</tr>
<tr>
<td>Overall Sample Size N</td>
</tr>
<tr>
<td>10,000</td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
</tr>
<tr>
<td>MSE</td>
</tr>
<tr>
<td>Variance</td>
</tr>
<tr>
<td>Bias</td>
</tr>
</tbody>
</table>

### Results for N-Skart Algorithm

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 − α</td>
<td>10,000</td>
<td>20,000</td>
</tr>
<tr>
<td>CI coverage</td>
<td>86.30%</td>
<td>88.10%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>59.64%</td>
<td>42.53%</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>1.3129</td>
<td>0.87047</td>
</tr>
<tr>
<td>CI coverage</td>
<td>92.00%</td>
<td>95.40%</td>
</tr>
<tr>
<td>Avg. CI half-length</td>
<td>77.28%</td>
<td>56.00%</td>
</tr>
<tr>
<td>Var. CI half-length</td>
<td>1.6728</td>
<td>1.1402</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Empirical Point-Estimator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Measures</td>
</tr>
<tr>
<td>Overall Sample Size N</td>
</tr>
<tr>
<td>10,000</td>
</tr>
<tr>
<td>Trunc. Sample Mean</td>
</tr>
<tr>
<td>MSE</td>
</tr>
<tr>
<td>Variance</td>
</tr>
<tr>
<td>Bias</td>
</tr>
</tbody>
</table>
Figure 3.25: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to ARTOP process (3.12) with $N = 10,000$
<table>
<thead>
<tr>
<th>N-Skart</th>
<th>MSER5 w/o correction</th>
<th>MSER5 with correction</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Graph" /></td>
<td><img src="image2.png" alt="Graph" /></td>
<td><img src="image3.png" alt="Graph" /></td>
</tr>
</tbody>
</table>

**Quantiles**

- **N-Skart**
  - 100.0% maximum: 6.27293
  - 99.5%: 2.96727
  - 97.5%: 2.43233
  - 90.0%: 2.18898
  - 75.0% Quartile: 2.01924
  - 50.0% Median: 1.87159
  - 25.0% Quartile: 1.743
  - 10.0%: 1.65233
  - 2.5%: 1.56118
  - 0.5%: 1.49504
  - 0.0% Minimum: 1.43277

- **MSER5 w/o correction**
  - 100.0% maximum: 8.94992
  - 99.5%: 4.65359
  - 97.5%: 2.14119
  - 90.0%: 1.83989
  - 75.0% Quartile: 1.60085
  - 50.0% Median: 1.2967
  - 25.0% Quartile: 1.11661
  - 10.0%: 1.05038
  - 2.5%: 1.01369
  - 0.5%: 1.0028
  - 0.0% Minimum: 1.00009

- **MSER5 with correction**
  - 100.0% maximum: 3.62304
  - 99.5%: 2.22755
  - 97.5%: 2.03869
  - 90.0%: 1.84003
  - 75.0% Quartile: 1.62721
  - 50.0% Median: 1.30336
  - 25.0% Quartile: 1.12149
  - 10.0%: 1.05495
  - 2.5%: 1.0171
  - 0.5%: 1.0028
  - 0.0% Minimum: 1.00034

**Moments**

- **N-Skart**
  - Mean: 1.9059716
  - Std Dev: 0.2699276
  - Std Err Mean: 0.0083441
  - Upper 95% Mean: 1.9225387
  - Lower 95% Mean: 1.8894075
  - N: 1000

- **MSER5 w/o correction**
  - Mean: 1.4107393
  - Std Dev: 0.4956022
  - Std Err Mean: 0.0156723
  - Upper 95% Mean: 1.4414938
  - Lower 95% Mean: 1.3798849
  - N: 1000

- **MSER5 with correction**
  - Mean: 1.3904157
  - Std Dev: 0.3118006
  - Std Err Mean: 0.00986
  - Upper 95% Mean: 1.4097644
  - Lower 95% Mean: 1.371067
  - N: 1000

Figure 3.26: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to ARTOP process (3.12) with $N = 20,000$
Figure 3.27: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to ARTOP process (3.12) with $N = 50,000$
Figure 3.28: Empirical distributions of truncated sample mean for N-Skart, MSER-5, and modified MSER-5 when applied to ARTOP process (3.12) with $N = 200,000$
The ARTOP process is characterized by a large positive initial transient. The mean of this process is 1.9091. From the results in Table 3.5, we can see that both the MSER-5 algorithms delivered unacceptable CI coverages. The coverages did not improve significantly for larger sample sizes. As against this, N-Skart delivered CI coverages that were close to the specified coverage levels.

While comparing the characteristics of the point estimator, it was observed that the MSER-5 algorithms considerably underestimated the steady-state mean, and the MSE delivered did not reduce significantly for larger sample sizes. Also, the bias was the main contributor to the MSE. For sample size $N=10,000$, the difference in the MSE delivered by all the methods was not significant. However, for larger sample sizes, the MSE obtained from N-Skart reduced by an order of magnitude for each subsequent increase in sample size.

The histograms for this test process indicate that the MSER-5 algorithms consistently underestimated the steady-state mean. This is also observed from the bimodal characteristics exhibited by the distribution of the truncated sample mean. As against this, N-Skart delivered point estimates that are centered close to the true mean of the ARTOP process and their variance was at least an order of magnitude smaller for larger sample sizes.

In summary, we concluded that in the ARTOP process, N-Skart significantly outperformed MSER-5 with respect to both point and CI estimators of the steady-state mean.
Chapter 4

Conclusions and Future Research

4.1. Conclusions

Through this research we have illustrated the simulation start-up problem and its significance in steady-state simulation output analysis. We have provided an overview of the various methods employed to address the problem of the initial transient. The most common technique used to address the start-up problem in practice is the replication-deletion approach in which an arbitrary set of initial observations is truncated from the given output sequence. The truncated sequence is referred to as the warm-up period of a steady-state output response. Various rules have been proposed in the literature that help us to determine the length of the warm-up period. Our research focussed on comparing the performance of N-Skart and MSER-5; two batch-means procedures designed to estimate the length of the warm-up period for the steady-state simulation output response provided and deliver a point and confidence interval (CI) estimate of the steady-state mean of the underlying output process.

When we considered the results delivered by N-Skart for all the test processes, we observed that it delivered estimates of the steady-state mean whose values were closely distributed about the true steady-state mean, as is evident from the histograms presented in the previous chapter. The empirical CI coverages delivered were in close conformance with the nominal coverage levels. It was observed that for test cases with pronounced positive bias and small sample sizes, N-Skart still produced valid CI estimates albeit with very large values of relative precision for the CI half-length (> 40%). This can be considered as a built-in safety feature for N-Skart which indicates that due to the pronounced bias in the test process, it was necessary to increase the sample size in order to achieve practically useful results. With a subsequent increase in sample size, the relative precision of the CI half-length delivered by N-Skart was significantly reduced. Also, in addition to addressing the problem of initial transient, N-Skart also made adjustments to the classical batch-means Student’s $t$-ratio to account for effects due to correlation and nonnormality in the output sequence. It is worth mentioning here that regardless of the test process used or the sample size considered, N-Skart successfully delivered a point estimate of
the steady-state mean that was approximately free of initialization bias and a CI estimate that was in close conformance with the nominal CI coverage level. All in all, we can say that N-Skart is a robust and efficient procedure that can be employed for addressing the simulation start-up problem.

Given a simulation output response of arbitrary length, the MSER-5 truncation heuristic is designed to estimate a truncation point such that the truncated sequence is approximately free of any initialization bias. The optimum MSER-5 truncation criterion minimizes the width of the CI about the truncated sample mean. This method is highly appealing intuitively and is much simpler to implement in practice as compared to N-Skart. However, from the results in the previous chapter it is evident that MSER-5 was not effective in addressing the problem of initial transient. The point estimates for the steady-state mean provided by MSER-5 exhibited considerable residual bias and variance. Also, the CI estimates delivered by MSER-5 had empirical CI coverages that did not conform to the nominal coverage levels. For nominal coverages of 90% and 95%, the empirical CI coverages delivered by MSER-5 for all test processes were observed to be < 40%. One of the grave shortcomings of the MSER-5 heuristic is that for a significant number of cases, it incorrectly estimates the truncation point to be at the end of the sample output sequence. The modification proposed to rectify this shortcoming of MSER-5 is an arbitrary increment to the minimum number of batches permissible for calculating the truncation criterion. There are no fixed guidelines regarding the choice of the increment size, which leaves its utility open to doubt. The results from the modified MSER-5 algorithm achieved some success in terms of reduced bias and variance in the estimate of the truncated sample mean, although the improvement in terms of empirical CI coverage levels was not significant. Also, the modified algorithm still estimated the optimal truncation point near the end of the output sequence for significant number of replications. Another lacuna in the MSER-5 heuristic is that while calculating the CI for the truncated sample mean, it assumes that the truncated sequence is i.i.d normal. Hence, it does not account for the skewness and autocorrelation in the output responses, as a result of which it delivers an extremely narrow CI. As a result of the above shortcomings, the distributions of the truncated sample means delivered by MSER-5 for almost all test processes were skewed and spread over a greater interval than corresponding distributions obtained from N-Skart’s results. Typically, for test processes with pronounced positive bias and small sample sizes, MSER-5 underestimated the value of the steady-state mean of the process and hence, the distributions of the truncated sample means were skewed. Moreover, for some test problems such as the ARTOP process, the distribution of the truncated sample mean delivered by MSER-5 exhibited bimodal characteristics, which raises serious questions about the effectiveness of this heuristic in estimating the truncated sample mean. All in all, we can say that although the MSER-5 truncation heuristic is highly intuitive and easy to implement, it does not deliver optimum results due to a few significant shortcomings.

Thus, considering the above discussion regarding the individual performances of N-Skart and MSER-5 we can conclude that N-Skart successfully outperformed MSER-5, for most of the test processes under consideration. N-Skart makes skillful use of the batch-means method to address the problems of cor-
relation and skewness (nonnormality) by incorporating provisions for flexible batch count and batch sizes in its algorithm. As against this, MSER-5 restricts the batch size to five, thus limiting its ability to adapt to different test processes. N-Skart provides greater utility to the user by delivering robust results that conform to user-specified sample size and nominal coverage levels. MSER-5 on the other hand, on several occasions, truncated the entire output sequence and delivered empirical CI coverages that were significantly less than the specified nominal coverages.

### 4.2. Contributions of the Current Research

In summary, the contributions of this research are as follows.

- We performed the first large-scale implementation and experimental evaluation of MSER-5 on a wide range of test processes.
- We discovered a minor revision of the formulation of N-Skart that improved its performance in most difficult test processes.
- We extended the experimental evaluation of N-Skart to include more difficult test processes—for example, $M/M/1$ queue-waiting times with 113 initial customers and 90% server utilization.
- We conducted the first large-scale performance comparison and analysis of N-Skart and MSER-5 in terms of:
  - Identification of anomalies in MSER-5 results; and
  - Analysis of N-Skart’s behavior when sample size $N$ is too small.

### 4.3. Directions for Future Research

On the basis of this research, the following recommendations have been made for future work:

- The main function of the MSER-5 heuristic is to deliver a truncated sample mean based on the truncation point estimated by the MSER-5 truncation criterion. Since it is desirable to have a valid CI associated with the point estimator of the steady-state mean, it would be helpful to combine the MSER-5 heuristic with a procedure that delivers a valid CI estimator.

- The MSER-5 heuristic is a special case of the MSER-$m$ heuristic which divides the given simulation output sequence into batches of size $m$. A fixed batch size limits the flexibility of this procedure to account for different degrees of correlation in various stochastic processes. It would be highly desirable to augment MSER-$m$ with an automatic procedure for determining an appropriate value of the batch size $m$ on each application of the procedure.
• Implementing N-Skart is much more difficult than implementing MSER-5. Simplified, computationally efficient versions of N-Skart should be implemented in portable, robust software that can be easily invoked "on the fly" in standard simulation environments or on a stand-alone basis.

• The experimental performance evaluation in the current research should be substantially expanded to include a much greater diversity of test processes with different types of transient behavior.
REFERENCES


APPENDICES
Figure 1: C Code for N-Skart

```c
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<stdlib.h>
#include "nr3.h"
#include "ran.h"
#include "gamma.h"
#include "incgammabeta.h"
#include <math.h>
#include<conio.h>
#include<stdio.h>

/*double norm(int *i,double *x1,double *x2)
*Function norm (generates normal deviates with mean 0 and standard deviation 1)
*#############################################################################

double skewfn
/* Calculates the skewness of an array between the elements n1 and n2 for the calculated mean and standard deviation
*#############################################################################

double varfn
/* Calculates the variance of an array between the elements n1 and n2 for the calculated mean
*#############################################################################

double meanfn
/* Calculates the mean of an array between the elements n1 and n2
*#############################################################################

float expdev22
/*Function expdev22 (generates exponentially distributed interarrival times with lambda = 0.9)
*#############################################################################

float expdev11
/*Function expdev11 (generates exponentially distributed service times with mu = 1)
*#############################################################################

double mean1
/* Calculates the mean of an array between the elements n1 and n2 for the calculated mean
*#############################################################################
```

69
/* Function find (calculates von Neumann test statistic and z. It is a part of the von Neumann randomness test) */
double fun2(double output, double *mean, double *var, double *pz, double *dstar, double *varl, double *varN)
{
    double temp, mean, var, k;
}

/* Function setdstar (calculates the maximum permissible value of d which is d*) */
double setdstar(void)
{
    double temp, mean, var, k;
    for (temp = mean; temp < k; temp++)
    {
        if (temp == k)
        {
            mean = temp;
        }
    }
}

/* Function fun2 (calculates ck: the von Neumann test statistic and z. It is a part of the von Neumann randomness test) */
double fun2(double output, double *mean, double *var, double *pz, double *dstar, double *varl, double *varN)
{
    double temp, mean, var, k;
    for (temp = mean; temp < k; temp++)
    {
        if (temp == k)
        {
            mean = temp;
        }
    }
}

/* Function find (calculates von Neumann test statistic and z. It is a part of the von Neumann randomness test) */
double fun2(double output, double *mean, double *var, double *pz, double *dstar, double *varl, double *varN)
{
    double temp, mean, var, k;
    for (temp = mean; temp < k; temp++)
    {
        if (temp == k)
        {
            mean = temp;
        }
    }
}
```plaintext
for (q=1;q<=qmax;q++)
    { 
        a1=q;     //dividing the initial sample into k nonspaced batches of size n each (part of step[1])
        a=q;
        s=q;
        /*-------------------------------*/
        /* Generate N1 random reals in (0, 1) */
        for (i=0; i<N1; i++)
            { 
                x1[i] = myranq1.doub();
                x2[i] = myranq2.doub();
            }
        /*-------------------------------*/
        /* Calculate the waiting times for ARTOP process */
        for(i=1;i<=N1;i++)
            { 
                if(i==1)
                    { 
                        int tmpj = i;
                        it = &tmpj;
                        zu1 = z.*99;
                    }
                else
                    { 
                        zu1 = zu;
                    }
                zu = (0.995*zu1)+(0.09987*norm(it,x1,x2));
                phi1 = (1/pow(2*M_PI,0.5))*(pow(M_E,-1*pow(zu,2)/2)); // M_PI gives pi and M_E gives e
                if(zu1 >= 0)
                    { 
                        t = (1/(1+(p*zu)));
                        phi2 = 1-(phi1*((b1*t)+(b2*pow(t,2))+(b3*pow(t,3))+(b4*pow(t,4))+(b5*pow(t,5))));
                    }
                else
                    { 
                        t = (1/(1-(p*zu)));
                        phi2 = (phi1*((b1*t)+(b2*pow(t,2))+(b3*pow(t,3))+(b4*pow(t,4))+(b5*pow(t,5))));
                    }
                output[i] = (1/(pow((1-phi2),(1/2.1))));
            }
        /*-------------------------------*/
        /* Calculate the waiting times for AR1 process */
        for(i=1;i<=N1;i++)
            { 
                if(i == 1)
                    { 
                        int tmpj = i;
                        it = &tmpj;
                    }
                output[i]=0;
            }
        for(i=2;i<=N1;i++)
            { 
                temp = (100+(0.995*(output[i-1]-100))+norm(it,x1,x2));
                if(temp > 0)
                    { 
                        output[i]= temp;
                    }
                else
                    { 
                        output[i] = 0;
                    }
            }
        /*-------------------------------*/
        /* Generate N1 random reals in (0, 1) with streams g1,g2 */
        for (i=1;i<=N1;i++)
            { 
                expdev1[i] = expdev11(x1[i]);
                expdev2[i] = expdev22(x2[i]);
            }
        /*-------------------------------*/
        /* Calculate the waiting times for M/M/1 queue */
        output[1]=0;
        for(i=2;i<=N1;i++)
            { 
                if(output[i-1]+expdev1[i-1]-expdev2[i-1] > expdev2[114])
                    output[i] = output[i-1]+expdev1[i-1]-expdev2[i-1];
                else
                    output[i] = 0;
            }
        /*-------------------------------*/
        /* Calculate the waiting times for M/M/1 loaded queue with 113 initial customers*/
        for(i=1;i<=113;i++)
            { 
                expdev2[i] = 0;
            }
        temp=0;
        for(i=1;i<=113;i++)
            { 
                temp=temp+expdev1[i];
                if(temp > expdev2[114])
```
\[ \text{output}[1] = \text{temp} - \expdev2[114]; \]

\[ \text{else} \]

\[ \text{output}[1] = 0; \]

\[ \text{for}(i=2; i\leq N1; i++) \]

\[ \text{if} (\text{output}[i-1] + \expdev1[i+112] - \expdev2[i+113] > 0) \]

\[ \text{output}[i] = \text{output}[i-1] + \expdev1[i+112] - \expdev2[i+113]; \]

\[ \text{else} \]

\[ \text{output}[i] = 0; \]

/*----------------------------------------------------------------------------*/
/* Print results to file*/
/*for(i=1; i\leq N1; i++)\]

\[ \text{if}(q == 1) \]

\[ \text{fprintf}(fp2, \"%f\n\", \text{output}[i]); \]

/*----------------------------------------------------------------------------*/
/* \[1\]-[1) Compute skewness of last 80% of simulation output readings*/
N
\[ = 0.8 * \text{N1}; \]

meaN
\[ = \text{mean}\{}(N1 - N, N1 + 1), \text{N1}, \text{output}\}; \]

varN
\[ = \text{var}\{}(N1 - N, N1 + 1), \text{N1}, \text{output}, \text{meaN}\}; \]

stdevN
\[ = \sqrt{\text{varN}}; \]

skewN
\[ = \text{skew}\{}(\text{output}); \]

/*----------------------------------------------------------------------------*/
/* \[1\]-[2) Compare skewness to finalize m, the initial batch size*/
//setting the no. of times batch count has been deflated in the randomness test (b) to 0
b
\[ = 0; \]

//setting current no. of batches in a spacer (d) to 0
\[ d = 0; \]

//setting maximum no. of batches in a spacer (d*) to 10
\[ d\star = 10; \]

//setting the spacer size (s) to 0
\[ s = 0; \]

\[ \text{if}(\text{fabs}(\text{skewN}) > 4) \]

\[ \text{temp} = \text{floor}(\text{N1}/1280); \]

\[ \text{if}(\text{temp} > 16) \]

\[ m = 16; \]

\[ \text{else} \]

\[ m = \text{temp}; \]

\[ \text{else} \]

\[ m = 1; \]

/*----------------------------------------------------------------------------*/
/* \[2\]-[a] Apply von Neumann’s Randomness test to current set of k batch means*/
/*\[2\]-[a] Apply von Neumann’s Randomness test to current set of k batch means*/
while(a1 == 0)

\[ a = 0; \]

setdstar(\text{output}, k, k1, m, c, z, \text{mean}, \text{mean1}, d, \text{n}, \text{s}, \text{meanl}, \text{varl}, \text{stdevl}, \text{skewl}, \text{dstar});

/*----------------------------------------------------------------------------*/
/* \[2\]-[d] Compare c with z*/
d0 = 0;

while(a == 0)

\[ \text{if}(\text{fabs(c)} < e) \]

\[ \text{if}(d < d0) \]

\[ \text{if}(d == 0) \]

\[ k1 = k; \]

\[ a = 1; \]

\[ a0 = 1; \]

\[ \text{printf}(\"\text{The batch size}(k) is } -1f\text{", s); \]

\[ \text{printf}(\"\text{The number of batches}(k) is } -1f\text{, } k1; \]

\[ \text{printf}(\"\text{The spacer size}(s) is } -1f\text{, } s; \]

\[ \text{printf}(\"\text{The total number observations (sample size) required}\text{ is } -1f\text{", s); \}

\[ d = 1; \]

\[ \text{else} \]

\[ \text{if}(d < d0) \]

\[ d0 = 0; \]

72
for(1;i<=k1;i++)
{    mean1[i] = 0;
    r=r+1;
    s=s+m;
    d=d+1;
    k1 = floor(n/(d+1)*m);
    for(1;i<=k1;i++)
    { if(i%r==0)
    {    mean1[i/r] = mean[i];
    }
    fun2(output,x1,k1,s,x1,x2,mean,mean1,k,d);
    }
    else
    { temp = sqrt(2);
      temp1 = ceil(temp*m)*ceil(0.9*k);
      if(temp1 <= N1)
      {    m = ceil(temp*m);
          k = ceil(0.9*k);
          n = k*m;
          b = b + 1;
          d = 0;
          
          if(d > dstar)
          { w = d;
            N2 = N1-w;
          }
          if(temp1<=N1)
          {    k1 = temp1;
          }
          else
          {    k1 = k;
          }
      }
      else
      {    c = 'y';
      }
    }
  }
  else
  { printf("Insufficient Data. Cannot proceed with current procedure");
    if(q == 1)
    {    scanf("%c",&c);
    }
    else
    {    printf("Do you want to continue to compute the confidence interval?\nplease enter your choice(y/n):\n");
      if(q == 1)
      {    scanf("%c",&c);
      }
    }
  }
  //printf("*************************************************************");
  //printf("\nInsufficient Data. Cannot proceed with current procedure\n");
  //printf("Do you want to continue to compute the confidence interval?\nplease enter your choice(y/n):\n");
  //printf("*************************************************************\n");
  //scanf("%c",&c);
  //if(q == 1)
  //{    scanf("%c",&c);
  //}
  //else
  //{
  }
}
/*von Neumann Randomness test completed*/
/*----------------------------------------------------------------------------*/
/*    Reinflate batch count, reset batch size*/
if(c == 'y')
{    printf("--------------------------------------\n");
    printf("\n\n\nStep 5[a] \n--------------------------------------\n");
    if(d > dstar)
    {    w = d;
        N2 = N1-w;
    }
    /*
     Set value of k1*/
    temp=ceil(k1*pow(1/0.9,k1));
    if(temp <= 1)
    {    k1=temp;
    }
    else
    {    k1=k1;
    }
    /*
     Calculate inflation factor*/
    f=sqrt(0.2/(k1*m));
    /*
     Set value of k1*/
    temp=floor(f*k1);
    if(temp <= 1024)
    {    k1=temp;
    }
    else
    {    k1=1024;
    }
    printf("\n\nFinal batch count(k') = %f\n");
  }
}
/** Set batch size m/**
if(k1 < 1024)
  m = floor(360);
else
  m = floor((360/320000000));
printf("sketch size(m) = %f",m);

/* Calculate truncated nonspaced batch means*/
for(j=1; j<k1; ++j)
  mean2[j]=meanfn((i1-((k1-1)+1)*m)+1,(i1-((k1-1)+1)*m)+1,input);

/* Update warmup period*/
var3g = varfn(i1,k1,mean2,mean3);

/* Calculate variance of truncated nonspaced batch means*/
var2g = varfn(i1,k1,mean2,mean3);

/* Calculate sample estimate of the lag one correlation of truncated nonspaced batch means*/
for(i=1; i<(k1-1); ++i)
  temp = temp+(mean2[i]+mean2[i+1]+mean3)*imax1,i=1;i+1;imax1)
  j = ((i-1)*i)/12w+1);  
  mean3[1]=mean2[1];

/* Calculate grand mean of spaced batch means*/
mean3g = meanfn(k2,1,k1,mean3);

/* Calculate variance for the spaced batch means*/
var3g = varfn(k2,1,k1,mean3,mean3g);
stdev3g = sqrt(var3g);

/* Calculate skewness for spaced batch means*/
for(i=1; i<k1; ++i)
  temp = temp+(mean3[i]+mean3[i]+mean3)+mean3[1]-mean3[1]);
  skew3g = ((k2-temp*/(k2-1)*(k2-2)));}/* Calculate k, beta, and N**
  temp = pow(var3g,1.5);  
  beta = pow(abs(beta));
  N = pow(abs(N));
  printf("k = %f",temp);  
  printf("beta = %f",beta);  
  printf("N = %f",N);
  if(abs(temp-1) < 0.000001){
    beta = -1;
    N = 1;
  }
  printf("*beta = %f",beta);  
  printf("*N = %f",N);  
  printf("k = %f",temp);  
  printf("beta = %f",beta);  
  printf("N = %f",N);  
  printf("*beta = %f",beta);  
  printf("*N = %f",N);
/* Write the observations to file*/
/*----------------------------------------------------------------------------*/
/* Calculate and print confidence interval*/
/*----------------------------------------------------------------------------*/

if (temp >= mean2g-hl1 && temp <= mean2g+hl2)
    temp = temp/mean2g;

printf("%d) [ %f , %f ]
The confidence interval for the steady state mean is:
phiHat = %f
sample variance = %f
sample mean = %f
spaced sample skewness = %f
spaced sample variance=%f
k'' = %f
d' = %f

%d)		%f		%f		%f		%f		[ %f , %f ]		%d		%f

/* if(100 >= mean2g-hl1 && 100 <= mean2g+hl2)
%d)		%f		%f		%f		%f		[ %f , %f ]		%d		%f

getch();
return 0;
Figure 2: C Code for MSER-5

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include "nr3.h"
#include "incgammabeta.h"
#include "gamma.h"
#include <string.h>

#define b5 1.330274429
#define b4 -1.821255978
#define b3 1.781477937
#define b2 -0.356563782
#define b1 0.319381530
#define p 0.2316419
#define N4 400000
#define N1 500000

int main()
{
  double x1 = (double)malloc(400000*sizeof(double));
  double x2 = (double)malloc(400000*sizeof(double));
  double output = (double)malloc(400000*sizeof(double));
  double mean = (double)malloc(200001*sizeof(double));
  double data = (double)malloc(400001*sizeof(double));
  double mean, t,norm ,statistic, meanfinal, hl, temp;
  double x1[100000], y[100000];
  Ranq1 ym1[1], ym2[1];
  FILE *fp = fopen("mser\mmempty_50000_90.nc.dat","a");
  // Generate N random walks in (0,1) /
  for (q=0; q<10000; q++)
  {
    for (i = 0; i < 10; i++)
    {
      x1[i] = ym1[1].double();
      x2[i] = ym2[1].double();
    }
    // ARTOP process /
    for(t=0; t<200001; t++)
    {
      if(t==10)
      {

    }
void AR1_process()
{
    int tmpj = i;
    it = &tmpj;
    zu1 = 3.4;
}
else
{
    zu1 = zu;
}
zu = ((0.995*zu1)+(0.09987*norm(it,x1,x2)));// M_PI gives pi and M_E gives e
phi1 = (1/pow(2*M_PI,0.5))*(pow(M_E,-1*pow(zu,2)/2));
if(zu1 >= 0)
{
    t = (1/(1+(p*zu)));
    phi2 = 1-(phi1*((b1*t)+(b2*pow(t,2))+(b3*pow(t,3))+(b4*pow(t,4))+(b5*pow(t,5))));
}
else
{
    t = (1/(1-(p*zu)));
    phi2 = (phi1*((b1*t)+(b2*pow(t,2))+(b3*pow(t,3))+(b4*pow(t,4))+(b5*pow(t,5))));
}
output[i] = (1/(pow((1-phi2),(1/2.1))));
}
/*----------------------------------------------------------------------------*/
/* AR1 process */
/*for(i=1;i<=N1;i++)
{
    if(i == 1)
    {
        int tmpj = i;
        it = &tmpj;
    }
    output[i]=0;
}
for(i=2;i<=N1;i++)
{
    temp = (100+(0.995*(output[i-1]-100))+norm(it,x1,x2));
    if(temp > 0)
    {
        output[i]= temp;
    }
    if(temp <= 0)
    {
        output[i] = 0;
    }
}
/*----------------------------------------------------------------------------*/
/* Generate exponentially distributed random numbers */
for (i = 1; i <=N1+112; i++)
{
    expdev1[i] = expdevv1[w1[i]];
    expdev2[i] = expdevv2[w2[i]];
}
/*----------------------------------------------------------------------------*/
/* Calculate the waiting-times for M/M/1 queue */
output[1]=0;
for(i=1;i<=N1;i++)
{
    if(output[i-1]+expdev1[i-1]-expdev2[i] > 0)
    {
        output[i]=output[i-1]+expdev1[i-1]-expdev2[i];
    }
    else
    {
        output[i] = 0;
    }
}
/*----------------------------------------------------------------------------*/
/* Calculate the waiting-times for M/M/1 loaded queue with 113 initial customers*/
/*for(i=1;i<=N1;i++)
{
    expdev2[i] = 0;
    temp=0;
    for(i=1;i<=113;i++)
    {
        temp=temp+expdev1[i];
    }
    if(temp > expdev2[i+113])
    {
        output[i] = temp - expdev2[i+113];
    }
    else
    {
        output[i] = 0;
    }
    for(i=2;i<=N1;i++)
    {
        if(output[i-1]+expdev1[i-1]-expdev2[i] > 0)
        {
            output[i] = output[i-1]+expdev1[i-1]-expdev2[i];
        }
    }
}
else
    output[i] = 0;
}
/*----------------------------------------------------------------------------*/
/* Implement MSER5*/
statistic = 1000;
N2 = (N1/5);
/* Divide output sequence into batches of size 5 and calculate batch means*/
for (i=N1;i>=1;i=--)
{
    temp = 0;
    for(j=i;j>=i-4;j--)
    {
        temp = temp + output[j];
    }
    mean[i/5] = (temp/5);
}
/* Calculate test statistic and find the minimum test statistic*/
temp = 0;
for (i=1;i<=(N2-1);i++)
{
    meannew = 0;
    temp = 0;
    for(j=N2;i=+)
    {
        temp = temp + mean[j];
    }
    meannew = (temp/(N2-i));
    temp = 0;
    for(j=i;j>i-5;j++)
    {
        temp = temp + pow(mean[j] - meannew, 2.0);
    }
    stat[i] = (temp/pow((N2-1)+i, 2));
    stat1[i] = (i*5)/sqrt(stat[i]);
    if(stat1[i] < statistic)
    {
        meanfinal = meannew;
        statistic = stat[i];
        trunc = ((i-5)*0.5);
        hl = stat[i];
        prechl = stat[i]/meanfinal;
        varhl = stat[i]*(N2-1-i);
    }
}
/*if(1.002 >= meanfinal-hl && 2.002 <= meanfinal+hl)
 * */
/*if(100 >= meanfinal-hl && 100 <= meanfinal+hl)
 * */
/*if(9 >= meanfinal-hl && 9 <= meanfinal+hl)
 * */
/*if(4 >= meanfinal-hl && 4 <= meanfinal+hl)
 * */
printf("%d) %f	%f	%f	%f	[%f,%f]	%d	%f
", q, meanfinal, hl, prechl, varhl, meanfinal-hl, meanfinal+hl, y, trunc);
fprintf(fp,"%d) %f,%f,%f,%f,[ %f,%f ] %d,%f
", q, meanfinal, hl, prechl, varhl, meanfinal-hl, meanfinal+hl, y, trunc);
getch();
}#*/
/*############################################################################*/