ON AUTOMATED SEQUENTIAL STEADY-STATE SIMULATION

A thesis
submitted in partial fulfilment
of the requirements for the degree
of
Doctor of Philosophy in Computer Science
in the
University of Canterbury
by
Jong-Suk Ruth Lee

University of Canterbury
December 2000

To my God who made this possible

Contents

ABSTRACT **1**

1 INTRODUCTION **3**

1.1 Stochastic Discrete-Event Simulation 3

1.2 Sequential Steady-State Simulation 8

1.2.1 Run-Length Control in Sequential Steady-State Simulations 10

1.3 Parallel and Distributed Simulation 13

1.3.1 Single Replication in Parallel 15

1.3.2 Multiple Replications in Parallel 20

1.4 Organisation of the Thesis 22

2 COVERAGE AS THE PERFORMANCE MEASURE OF SEQUENTIAL STEADY-STATE SIMULATION **25**

2.1 Introduction 25

2.2 Interval Estimators for Coverage Analysis 30

2.2.1 Interval Estimator Based on the Normal Distribution . . 31

2.2.2 Interval Estimator Based on the *Arcsin* Transformation . 32

2.2.3 Interval Estimator Based on the F Distribution 33

2.3 Performance Evaluation of Three Interval Estimators 34

2.4 Comparisons of Three Interval Estimators with Exact Values . . 38

2.5 Rules for Experimental Coverage Analysis of Sequential Simulation 42

2.5.1 Rules for Experimental Coverage Analysis 44

2.5.2 Experimental Results 46

2.6 Conclusions 51

3 A PROBLEM OF TOO SHORT RUNS IN SEQUENTIAL STEADY-STATE SIMULATION 53

3.1 Introduction 53

3.2 A Problem of Early Stopping: Experimental Evidence 55

3.3 Heuristic Rules for Preventing the Final Results Coming from 'Too Short' Runs 66

3.4 Performance Evaluation of the Proposed Heuristic Rules 71

3.5 Theoretical and Experimental Run-Length 90

3.6 Relationship Between Coverage and Run-Length 92

3.7 Conclusions 96

4 A PROBLEM OF TOO SHORT SEQUENTIAL STEADY-STATE REGENERATIVE SIMULATIONS OF MEAN VALUES 99

4.1 Introduction 99

4.2 Ratio Estimators for Use in the Sequential Method of RCs . . . 101

4.3 Sequential Method of RCs: A Problem of Early Stopping 106

4.4 Location of the First Checkpoint for the Sequential Method of RCs 114

4.5 Coverage Analysis for Sequential Method of RCs 123

4.6 Conclusions 127

5 AUTOMATION OF SEQUENTIAL STEADY-STATE QUANTILE ESTIMATION 129

5.1 Quantiles: Importance and Limitations 129

5.2 Sequential QE Based on the Method of RCs 132

5.2.1 Sequential QE Using the *Linear* Approach 133

5.2.2 Sequential QE Using the *Batching* Approach 135

5.3 Sequential QE for Non-Regenerative Processes 138

5.3.1 Detection Methods of the Initial Transient Period for QE 139

5.3.2 Comparisons of Two Statistical Tests for QE 143

5.3.3 Sequential QE Using the *Spectral* P^2 Approach 149

5.4 Experimental Coverage Analysis for Sequential QE 153

5.5 Performance Evaluation of Sequential QE Approaches 156

5.6 Conclusions 170

6 SPEEDUP IN PARALLEL AND DISTRIBUTED SIMULATION 173

6.1 Introduction 173

6.2 Theoretical Speedup in the MRIP Scenario 175

6.3 Empirical Speedup in the MRIP Scenario 180

6.4 Conclusions 188

7 SUMMARY, CONCLUSIONS, AND FUTURE RESEARCH191

7.1 Summary and Conclusions 191

7.2 Suggestions for Future Research 197

REFERENCES 200

APPENDICES 221

A Automated Simulation Package: Akaroa-2 223

A.1 Architecture of Akaroa-2 223

A.2 Transient Period Detection in Akaroa-2 225

A.3 Random Number Generator in Akaroa-2 227

A.3.1 The Multipliers used in Akaroa-2 232

B Selected Methods of Sequential Simulation Output Data Analysis of Mean Values 233

B.1 Batch Means Methods 236

B.2 Methods Based on Spectral Analysis 238

B.3 Regenerative Cycle Method 241

C Standardised Time Series Used in Statistical Tests for Detection of the Initial Transient Period 249

D Sequential Procedures for QE 253

D.1 Sequential QE Using the *Linear* Approach 253

D.2 Sequential QE Using the *Batching* Approach 257

D.3 Sequential QE Using the *Spectral* P^2 Approach 260

E Statistics from a Survey of Literature on Applications of Stochastic Simulation 267

F Theoretically Required Simulation Run-Length for Some Stationary Queueing Systems 271

F.1 Run-Length for the $M/M/1/\infty$ Queueing System 272

F.2 Run-Length for the $M/D/1/\infty$ Queueing System 275

F.3 Run-Length for the $M/H_2/1/\infty$ Queueing System 277

F.4 Theoretically Required Run-Length When Estimating the Mean Response Time 278

ACKNOWLEDGEMENTS 281

List of Figures

1.1 Proportions of all surveyed research papers reporting the results obtained by a stochastic simulation (average proportion is 51.45%) 6

1.2 Proportions of all papers based on a stochastic simulation, in which results were analysed statistically (average proportion is 23.55%) 7

1.3 Single Replication In Parallel ($P = n$ processors, M = M1 $\cup$ M2 $\cup \cdots \cup$ Mn) 16

1.4 Static load balancing (taken from [3]) 19

1.5 Dynamic load balancing (taken from [3]) 19

1.6 Multiple Replications In Parallel ($P = n$ processors) 20

2.1 Coverage analysis of the method of non-overlapping batch means (simulation of the $M/M/1/\infty$ queueing system) 27

2.2 Valid and invalid CIs in coverage analysis 31

2.3 CIs of coverage of SA/HW using the normal distribution, the *arcsin* transformation, and the F distribution 36

2.4 The CIs of proportions using the normal distribution, the *arcsin* transformation, and the F distribution ($\alpha = 0.01$ & $n = 20$) . . 38

2.5 Convergence of coverage ($M/M/1/\infty$ at $\rho = 0.9$, confidence level of 0.95) 43

2.6 Coverage convergence after discarding ‘too short’ simulation runs. Filtering started when at least 200 runs having invalid CIs were collected ($M/M/1/\infty$, load = 0.9, confidence level of 0.95) 45

2.7 Sequential coverage analysis using F distribution ($M/M/1/\infty$) . 48

2.8 Sequential coverage analysis using F distribution ($M/D/1/\infty$) . 49

2.9 Sequential coverage analysis using F distribution ($M/H_2/1/\infty$) 50

3.1 Convergence on a relative statistical error of 5% for the sequential method of NOBM (when estimating the mean response time in the $M/M/1/\infty$ queueing system at load = 0.9, checkpoints spaced linearly: 1,250 observations between two checkpoints) . . 54

3.2 Histogram of simulation run-lengths ($M/M/1/\infty$, load = 0.9, 10,000 replications) 57

3.3 Histogram of simulation run-lengths ($M/D/1/\infty$, load = 0.9, 10,000 replications) 58

3.4 Histogram of simulation run-lengths ($M/H_2/1/\infty$, load = 0.9, 10,000 replications) 59

3.5 Coverage of the CIs with Rule I (take the longest of R replications; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/M/1/\infty$ queueing system 73

3.6 Coverage of the CIs with Rule I (take the longest of R replications; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/D/1/\infty$ queueing system 74

3.7 Coverage of the CIs with Rule I (take the longest of R replications; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/H_2/1/\infty$ queueing system 75

3.8 Coverage of the CIs with Rule II (take the most accurate result out of R results obtained; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/M/1/\infty$ queueing system 77

3.9 Coverage of the CIs with Rule II (take the most accurate result out of R results obtained; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/D/1/\infty$ queueing system 78

3.10 Coverage of the CIs with Rule II (take the most accurate result out of R results obtained; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/H_2/1/\infty$ queueing system 79

3.11 Coverage of the CIs with Rule III (take the widest CIs of R replications; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/M/1/\infty$ queueing system 80

3.12 Coverage of the CIs with Rule III (take the widest CIs of R replications; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/D/1/\infty$ queueing system 81

3.13 Coverage of the CIs with Rule III (take the widest CIs of R replications; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/H_2/1/\infty$ queueing system 82

3.14 Coverage of the CIs with Rule IV (combining R replications; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/M/1/\infty$ queueing system 84

3.15 Coverage of the CIs with Rule IV (combining R replications; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/D/1/\infty$ queueing system 85

3.16 Coverage of the CIs with Rule IV (combining R replications; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/H_2/1/\infty$ queueing system 86

3.17 Coverage of the CIs with Rule V (combination of Rules III and IV). Estimation of the mean response time in the $M/M/1/\infty$ queueing system 87

3.18 Coverage of the CIs with Rule V (combination of Rules III and IV). Estimation of the mean response time in the $M/D/1/\infty$ queueing system 88

3.19 Coverage of the CIs with Rule V (combination of Rules III and IV). Estimation of the mean response time in the $M/H_2/1/\infty$ queueing system 89

3.20 Mean run-length of 10,000 independent simulation runs of estimating the mean response time 91

3.21 Coverage over run-lengths with Rule I (at load 0.9, $M/M/1/\infty$). T: theoretical requirement, S: experimental results of a single run, L2: experimental results of a longer run of 2, L3: experimental results of the longest run of 3, and L5: experimental results of the longest run of 5 93

3.22 Coverage over run-lengths with Rule I (at load 0.9, $M/D/1/\infty$). T: theoretical requirement, S: experimental results of a single run, L2: experimental results of a longer run of 2, L3: experimental results of the longest run of 3, and L5: experimental results of the longest run of 5 94

3.23 Coverage over run-lengths with Rule I (at load 0.9, $M/H_2/1/\infty$). T: theoretical requirement, S: experimental results of a single run, L2: experimental results of a longer run of 2, L3: experimental results of the longest run of 3, and L5: experimental results of the longest run of 5 95

4.1 Ranges and numbers of ‘extremely short’ simulation runs observed in the sequential method of RCs when estimating the mean response time at a confidence level of 0.95 with a statistical error $\leq 10\%$ 108

4.2 Distributions of simulation run-lengths, measured by the number of RCs, for the sequential method of RCs using different estimators (when estimating the mean response time at $\rho = 0.9$ from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq 10\%$) 112

4.3 Simulation run-lengths, measured by the number of RCs, with different locations of the first checkpoint ($M/M/1/\infty$, ρ=0.9, n_1 is the location of the first checkpoint) 115

4.4 Simulation run-lengths, measured by the number of RCs, with different locations of the first checkpoint ($M/D/1/\infty$, ρ=0.9, n_1 is the location of the first checkpoint) 116

4.5 Simulation run-lengths, measured by the number of RCs, with different locations of the first checkpoint ($M/H2/1/\infty$, ρ=0.9, n_1 is the location of the first checkpoint) 117

4.6 Coverage analysis vs different locations of the first checkpoint $n_1 = 2$, 10, 30, 50, 100, and 150 RCs (the confidence level = 0.95)118

4.7 Convergence of the coverage of the sequential method of RCs (when estimating the mean response time at $\rho = 0.5$ from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq 10\%$) 122

4.8 Convergence of the coverage of a sequential method of RCs when applying the location of the first checkpoint as having 100 RCs ($n_1 = 100$), collecting at least 200 bad CIs before stopping, and discarding 'too short' runs (when estimating the mean response time at $\rho = 0.5$ from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq 10\%$) 123

4.9 Coverage analysis of sequential method of RCs when applying the location of the first checkpoint as having 2 RCs ($n_1 = 2$) (non-sequential coverage analysis of 200 replications and sequential coverage analysis) 125

4.10 Sequential coverage analysis of sequential method of RCs when applying the two different locations of the first checkpoint as having 2 RCs ($n_1 = 2$) and 100 RCs ($n_1 = 100$) 126

5.1 Flowchart for the sequential *linear* QE in the method of RCs . . 134

5.2 Flowchart for the sequential *batching* QE in the method of RCs 136

5.3 Theoretical convergence of the cumulative distribution function of waiting times of the n-th customer in the $M/M/1/\infty$ queue to the theoretical steady-state (at a traffic intensity of $\rho = 0.8$) . 144

5.4 Comparisons of theoretical and empirical values of cumulative distribution function of waiting times in the queue when using the Schruben test for detecting an initial transient period ($M/M/1/\infty$ queueing system at a traffic intensity of $\rho = 0.8$) . 145

5.5 Comparisons of theoretical and empirical values of cumulative distribution function of waiting times in the queue when using the GSS test for detecting an initial transient period ($M/M/1/\infty$ queueing system at a traffic intensity of $\rho = 0.8$) 147

5.6 Flowchart for the method of sequential *spectral* P^2 QE in the method of non-RCs 151

5.7 Convergence of the coverage for QE ($M/M/1/\infty$ at $\rho = 0.5$ and 0.9 quantile, confidence level of 0.95) 154

5.8 Convergence of the coverage for QE after discarding 'too short' simulation runs ($M/M/1/\infty$ at $\rho = 0.5$ and 0.9 quantile, confidence level of 0.95) 155

5.9 Simulation run-lengths, measured by the number of RCs, for *linear* QE ($M/M/1/\infty$, 0.9 quantile, the location of the first checkpoint: $n_1 = 100$ RCs, 21 grid points spaced 0.2 units apart) 157

5.10 Coverage analysis of the *linear* QE approach in the sequential method of RCs ($M/M/1/\infty$, 0.9 quantile, *: the location of the first checkpoint, o: the number of grid points, and ♢: the space between grid points) 158

5.11 *Linear* QE approach in the method of RCs ($M/M/1/\infty$, ρ=0.1, 0.9 quantile, the location of the first checkpoint: $n_1 = 1,000$ RCs, 21 grid points spaced 0.2 units apart) 159

5.12 Simulation run-lengths, measured by the number of batches, for *batching* QE ($M/M/1/\infty$, 0.9 quantile, the location of the first checkpoint: $r_1 = 10$ batches, the batch size: 100 RCs) 161

5.13 Coverage analysis of the *batching* QE approach in the sequential RCs ($M/M/1/\infty$, 0.9 quantile, *: the location of the first checkpoint, and o: the batch size) 162

5.14 *Batching* QE approach in the sequential RCs ($M/M/1/\infty$, ρ=0.9, 0.9 quantile, the location of the first checkpoint: $r_1 = 10$ batches, the batch size: 100 RCs) 163

5.15 Coverage analysis of the sequential *spectral* P^2 QE approach ($M/M/1/\infty$, 0.9 quantile) 163

5.16 *Spectral* P^2 QE approach in the method of non-RCs ($M/M/1/\infty$, ρ=0.9, 0.9 quantile) 164

5.17 Bias of the three QE approaches: the *linear* QE, the *batching* QE, and the *spectral* P^2 QE ($M/M/1/\infty$, 0.9 quantile) 166

5.18 Coverage of the CIs with Rules I and V (proposed in Chapter 3) of the *linear* QE in the sequential method of RCs (when estimating the 0.9 quantile of response times in the $M/M/1/\infty$ queueing system, the location of the first checkpoint: $n_1 = 1,000$ RCs, the number of grid points: 31, and the space between grid points: 0.1) 167

5.19 Coverage of the CIs with Rules I and V (proposed in Chapter 3) of the *batching* QE in the sequential method of RCs (when estimating the 0.9 quantile of response times in the $M/M/1/\infty$ queueing system, the location of the first checkpoint: $r_1 = 10$ batches, and the batch size: 200 RCs per batch) 168

5.20 Coverage of the CIs with Rules I and V (proposed in Chapter 3) of the *spectral* P^2 QE approach in the sequential method of non-RCs (when estimating the 0.9 quantile of response times in the $M/M/1/\infty$ queueing system) 169

6.1 Ideal parallelism (taken from [58]) 176

6.2 Parallelism with two portions: one executes in parallel and the other can only execute sequentially (taken from [58]) 176

6.3 Speedup achievable with Equation (6.4) based on *Amdahl's law* 177

6.4 Speedup achievable theoretically under the MRIP scenario according to the truncated Amdahl's law ($P_{min} = 10$) 180

6.5 Speedup obtained under the MRIP scenario for the SA/HW method when estimating steady-state mean response times ($\rho = 0.5$) 182

6.6 Speedup obtained under the MRIP scenario for the method of RCs when estimating steady-state mean response times ($\rho = 0.5$)184

6.7 Speedup achieved from *spectral* P^2 QE based on the method of non-RCs to estimate the 0.9 quantile for the response time in the $M/M/1/\infty$ at a traffic intensity of $\rho = 0.5$ 186

6.8 Speedup achieved from *batching* QE based on the method of RCs to estimate the 0.9 quantile for the response time in the $M/M/1/\infty$ at a traffic intensity of $\rho = 0.5$ 187

A.1 Architecture of Akaroa-2 (taken from [28]) 224

A.2 Sequential coverage analysis using the method of SA/HW when estimating the mean response time in the $M/H_2/1/\infty$ queueing system with the two different PRNGs: Multiplicative Linear Congruential (MLC)-PRNGs and Combined Multiple Recursive LC-PRNGs (the confidence level = 0.95) 230

B.1 Flowchart for the sequential method of RCs 245

E.1 Statistics of research papers, published in the Proceedings of IEEE INFOCOM, in which results were obtained by simulation 268

E.2 Statistics of research papers, published in IEEE Transactions on Communications, in which results were obtained by simulation268

E.3 Statistics of research papers, published in IEEE/ACM Transactions on Networking, in which results were obtained by simulation269

E.4 Statistics of research papers, published in Performance Evaluation: An International Journal, in which results were obtained by simulation 269

List of Tables

1.1 Comparisons of some simulation packages 14

2.1 Coverage and its CIs of SA/HW using the normal distribution, the *Arcsin* transformation, and the F distribution (when estimating the mean response time at a confidence level = 0.95 with a statistical error $\leq$ 5%) 37

2.2 Upper limits of CIs of proportions ($\alpha = 0.01$ & $n = 20$) 39

2.3 Relative inaccuracy of upper confidence limits of proportions ($\alpha = 0.01$ & $n = 20$) 40

2.4 Invalid CIs of proportions using the normal distribution 41

3.1 Mean run-lengths of 10,000 sequential steady-state simulations (when estimating the mean response time at load = 0.9 with a relative statistical error of 10% at a confidence level = 0.95) . . 56

3.2 Sequential method of NOBM from 10,000 replications (when estimating the mean response time from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%) 61

3.3 Sequential method of SA/HW from 10,000 replications (when estimating the mean response time from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%) 61

3.4 Sequential method of RCs from 10,000 replications (when estimating the mean response time from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%) 62

3.5 Sequential method of NOBM from 10,000 replications (when estimating the mean response time from the $M/D/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%) 62

3.6 Sequential method of SA/HW from 10,000 replications (when estimating the mean response time from the $M/D/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%) 63

3.7 Sequential method of RCs from 10,000 replications (when estimating the mean response time from the $M/D/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%) 63

3.8 Sequential method of NOBM from 10,000 replications (when estimating the mean response time from the $M/H_2/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%) 64

3.9 Sequential method of SA/HW from 10,000 replications (when estimating the mean response time from the $M/H_2/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%) 64

3.10 Sequential method of RCs from 10,000 replications (when estimating the mean response time from the $M/H_2/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%) 65

3.11 The probability of R independent replications belonging to the class of 'too short' simulation runs (theoretical confidence level = 0.95) 68

4.1 Statistics from 10,000 replications using the sequential method of RCs with $n_1 = 2$ as the location of the first checkpoint (when estimating the mean response time from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%) 109

4.2 Statistics from 10,000 replications using the sequential method of RCs with $n_1 = 2$ as the location of the first checkpoint (when estimating the mean response time from the $M/D/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%) 110

4.3 Statistics from 10,000 replications using the sequential method of RCs with $n_1 = 2$ as the location of the first checkpoint (when estimating the mean response time from the $M/H_2/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%) 111

4.4 Statistics obtained from 3,000 replications for the sequential method of RCs with different estimators (when estimating the mean response time at $\rho = 0.9$ from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%) 113

4.5 Sequential method of RCs with the assumption of a minimum number of 100 RCs ($n_1 = 100$) collected before stopping, from 10,000 simulation replications: $M/M/1/\infty$, theoretical confidence level = 0.95 120

4.6 Sequential method of RCs with the assumption of a minimum number of 100 RCs ($n_1 = 100$) collected before stopping, from 10,000 simulation replications: $M/D/1/\infty$, theoretical confidence level = 0.95 120

4.7 Sequential method of RCs with the assumption of a minimum number of 100 RCs ($n_1 = 100$) collected before stopping, from 10,000 simulation replications: $M/H_2/1/\infty$, theoretical confidence level = 0.95 121

5.1 Statistics obtained from the Schruben and GSS tests for detecting the initial transient period in the steady-state estimation of means ($M/M/1/\infty$ queueing system at the 0.95 confidence level with the required relative statistical error of 10% or less) 148

5.2 The mean length of the initial transient periods detected by Schruben test when estimating 0.9 quantiles using the *spectral* P^2 QE approach from 6,000 independent replications for the $M/M/1/\infty$ queueing system 164

5.3 Means and CIs of 0.9 quantiles obtained from 6,000 independent simulation replications executed for the three QE approaches: *linear* QE, *batching* QE, and *spectral* P^2 QE, in the $M/M/1/\infty$ queueing system at a confidence level of 0.95 ($\surd$ means that the CIs of a quantile contain the theoretical quantile) 165

6.1 Statistics averaged from 100 independent sequential steady-state simulation runs applying the SA/HW method to estimate the mean response time at a traffic intensity of $\rho = 0.5$ with a relative statistical error of at least 5% at the 0.95 confidence level . 181

6.2 Statistics averaged from 100 independent sequential steady-state simulation runs applying the RCs method to estimate the mean response time at a traffic intensity $\rho = 0.5$ with a relative statistical error of at least 5% at the 0.95 confidence level 185

6.3 Statistics averaged from 100 independent sequential steady-state simulation runs applying the method of *spectral* P^2 QE to estimate the 0.9 quantile for the response time in the $M/M/1/\infty$ at a traffic intensity of $\rho = 0.5$ 186

A.1 Statistical test of one way ANOVA for the coverage obtained using the method of SA/HW when estimating the mean response time in the $M/H_2/1/\infty$ queueing system with the Multiplicative Linear Congruential (MLC)-PRNGs and Combined Multiple Recursive LC-PRNGs 231

E.1 Statistics of research papers published in the technical literature 270

F.1 Required run-length when estimating the mean waiting time in the $M/M/1/\infty$ queueing system at a 95% CI 275

F.2 Required run-length when estimating the mean waiting time in the $M/D/1/\infty$ queueing system at a 95% CI 277

F.3 Required run-length when estimating the mean waiting time in the $M/H_2/1/\infty$ queueing system at a 95% CI 279

F.4 Required run-length when estimating the mean response time from the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems 279

ON AUTOMATED SEQUENTIAL STEADY-STATE SIMULATION

Jong-Suk Ruth Lee
Department of Computer Science
University of Canterbury
Christchurch, New Zealand

ABSTRACT

The credibility of the final results from stochastic simulation has had limited discussion in the simulation literature so far. However, it is important that the final results from any simulations be credible. To achieve this, validation, which determines whether the conceptual simulation model is an accurate representation of the system under study, has to be done carefully. Additionally, a proper statistical analysis of simulation output data, including a confidence interval or other assessment of statistical errors, has to be conducted before any valid inferences or conclusions about the performance of simulated dynamic systems, such as for example telecommunication networks, are made.

There are many other issues, such as choice of a good pseudo-random number generator, elimination of initialisation bias in steady-state simulations, and consideration of autocorrelations in collected observations, which have to be appropriately addressed for the final results to be credible. However, many of these issues are not trivial, particularly for simulation users who may not be experts in these areas.

As a consequence, a fully-automated simulation package, which can control all important aspects of stochastic simulation, is needed. This dissertation focuses on the following contributions to such a package for steady-state simulation: properties of confidence intervals (CIs) used in coverage analysis, heuristic rules for improving the coverage of the final CIs in practical applications, automated sequential analysis of mean values by the method of regenerative cycles, automatic detection of the initial transient period for steady-state quantile estimation, and sequential steady-state quantile estimation with the automated detection of the length of initial transient period.

One difficulty in obtaining precise estimates of a system using stochastic simulation can be the cost of the computing time needed to collect the large amount of output data required. Indeed there are situations, such as estimation of rare events, where, even assuming an appropriate statistical analysis procedure is available, the cost of collecting the number of observations needed by the analysis procedure can be prohibitively large. Fortunately, inexpensive computer network resources enable computationally intensive simulations by allowing us to run parallel and distributed simulations. Therefore, where possible, we extend the contributions to the distributed stochastic simulation scenario known as the Multiple Replications In Parallel (MRIP), in which multiple processors run their own independent replications of the simulated system but cooperate with central analysers that collect data to estimate the final results.

Chapter 1

INTRODUCTION

1.1 Stochastic Discrete-Event Simulation

Discrete-event stochastic dynamical systems, such as those which can be modelled by queueing networks, occur in all areas of industry and business, including manufacturing processes, communication networks, and computer systems. They are often difficult to evaluate analytically, even when they are only moderately complex, due to their nonlinear behaviour. However, significant achievements in electronic and computer engineering have led to a proliferation of powerful computers in almost every office and business, and remarkable achievements in software technology have allowed very simple and efficient human-computer interfaces. These two developments have led to computer-based stochastic simulation becoming the most commonly and widely used tool for performance evaluation studies when analytical techniques do not suffice. Computer simulation has also been adopted for scientific investigations, in addition to the traditional theoretical and experimental studies.

Furthermore, the emergence of the world-wide web (WWW) has affected many areas including computer simulations. This phenomenon has introduced the (relatively) new concept of *web-based simulation* which represents a convergence of computer simulation methodologies and applications within the

WWW [35], [125]. This is now one of hot research topics for both simulation researchers and simulation practitioners. This will lead to computer-based stochastic simulation being an even more powerful tool for many disciplines [124].

It is essential to use a *valid simulation model* for any performance evaluation studies based on stochastic simulation. General guidelines on building valid simulation models can be found, for example, in [87] and [93]. However, the validity of the model is only the first step towards the credibility of the final results of any simulation study, since as Law and McComas wrote that *"the modelling phase of a system's simulation consumes only 30 - 40% of the total effort in most successful simulation projects"* [94]. Nevertheless, a great deal of time and money in simulation studies is spent mostly on model development and programming rather than over all the steps, which can be found, for example in [93] and [94], involved.

Warnings regarding the misuse of stochastic simulation as a performance evaluation tool of complex dynamic systems can be found, for example, in [44] and [83]. The misuse of stochastic simulation has led to a deep credibility crisis in the use of simulation studies for performance evaluation. Although the credibility of the final simulation results has hardly been discussed in the literature so far, it is probably as important as the problem of validation, which determines whether the conceptual simulation model is an accurate representation of a system under study.

In practice a common mode of application of simulations is to make a single simulation run of a somewhat arbitrary length and then to treat the resulting simulation estimates as the 'true' system's characteristics. Since random samples from various probability distributions are used to drive a simulation model, any output estimates are simply particular realizations of random variables that may have large variances. As a result, the estimates in a particular simulation run can differ greatly from the corresponding true values. The net effect of this approach is a significant probability of making erroneous conclusions about the performance of the system under study.

Following the scientific method ([175] and [179]), one should draw conclusions only from controlled and repeatable simulation experiments. This is necessary, for example, to facilitate comparisons between alternative systems, when many simulation runs of alternative systems with the same pseudo-random numbers may be required.

As discussed before, any stochastic simulation could be regarded as a statistical experiment, since the input processes driving a given simulation are random. Hence, a proper statistical analysis of simulation output data, including a confidence interval (CI) or probability statement, has to be undertaken before any valid inferences or conclusions about the performance of the investigated computer systems or telecommunication networks are made. Unfortunately, there is a reason why simulation output data analyses have often not been conducted in an appropriate manner.

The reason for inadequate analysis is that the output processes of many simulations are non-stationary and/or autocorrelated. Thus, classical statistical analysis techniques developed from independent and identically distributed observations are then not applicable to the analysis of such simulation output data. Some problems of simulation output data analysis, such as an initial transient period detection in steady-state simulation, and handling autocorrelations between collected observations, have no completely accepted solutions, and choosing the appropriate method to apply in simulation practice is often not easy.

Applying inadequate methods for analysing simulation output data has led to an alarming situation in all fields of performance evaluation, including telecommunication networks. The credibility of many research publications based on simulation studies can be questioned. We have conducted a survey of 2245 research papers published recently in Proceedings of INFOCOM (an annual IEEE International Conference on Computer Communications) from 1992 to 1998 (papers per year range between 156 and 177), IEEE Transactions on Communications from 1996 to 1998 (230, 227 and 221 papers), IEEE/ACM Transactions on Networking from 1996 to 1998 (83, 80 and 68), and Performance Evaluation Volumes 25 - 34 from 1996 to 1998. The survey shows that stochastic simulation is a preeminent tool of scientists and engineers working on performance evaluation of telecommunication networks, computer systems, and other similar systems [132]. Figure 1.1 shows the data obtained from that survey.

Our results also show that in about 76% of the surveyed papers the authors were not concerned with the random nature of the experimental results they obtained from their stochastic simulation studies; see Figure 1.2. This included papers simply reporting the average results (say, average over an arbitrary number of replications), with an unspecified statistical error. The majority of researchers do not mention whether their final simulation results have been subjected to an appropriate statistical analysis. Certainly, this cannot be an acceptable practice!

It would appear that one cannot rely on the majority of published results of performance evaluation studies of dynamic systems based on a stochastic simulation, since the final results lack credibility if an appropriate statistical analysis is not done. Other aspects of the credibility crisis are discussed, for example, in [129], [132], and [171]. Detailed results of the survey can be found in Appendix E.

There are many other issues, such as verification of the simulation program,

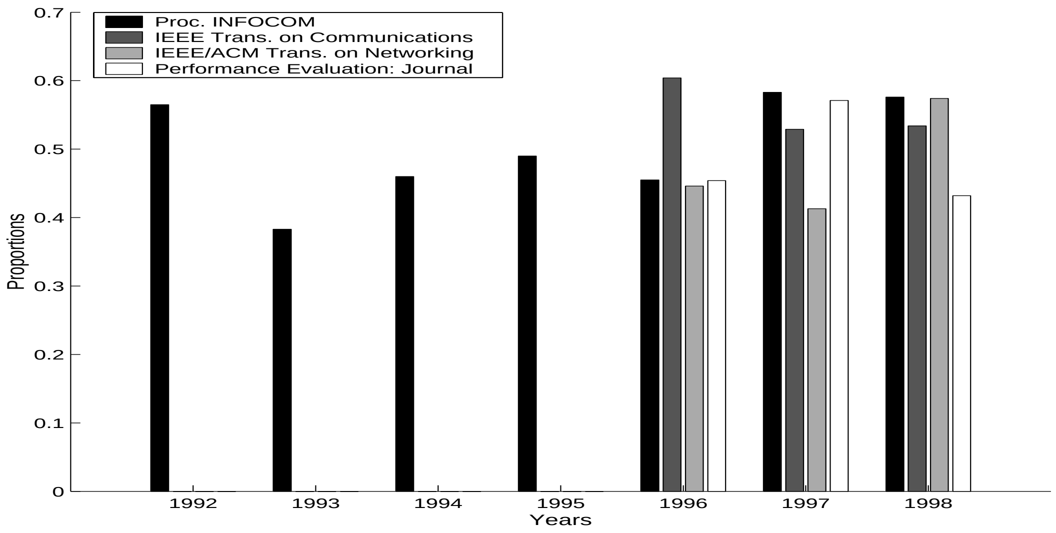


Figure 1.1: Proportions of all surveyed research papers reporting the results obtained by a stochastic simulation (average proportion is 51.45%)

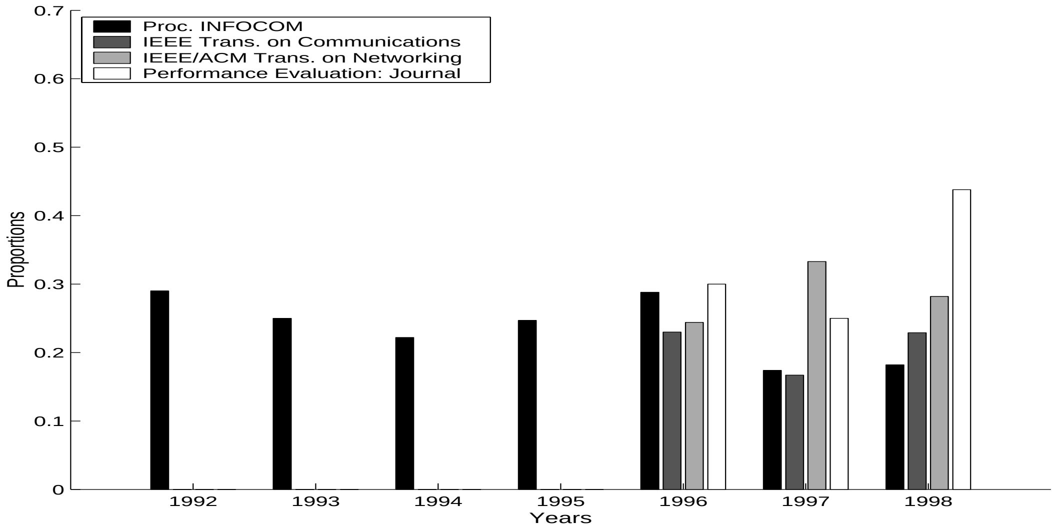


Figure 1.2: Proportions of all papers based on a stochastic simulation, in which results were analysed statistically (average proportion is 23.55%)

choice of a good pseudo-random number generator, elimination of initialisation bias in steady-state simulations, and consideration of autocorrelations in collected observations, which have to be seriously considered to achieve credibility of the final simulation results. However, many of these issues are not trivial, particularly for simulation users who are not expert in these areas. Thus, achieving a successful simulation result is difficult.

Consequently, a fully-automated simulation package, which can control and validate all aspects of a stochastic steady-state simulation, would be valuable. This dissertation focuses on the following contributions to such a package: CI estimations for coverage analysis, heuristic rules for improving the coverage of the final CIs in practical applications, the automated sequential analysis of mean values by means of regenerative cycles (RCs), automatic detection of the initial transient period for steady-state quantile estimation, and automated sequential steady-state quantile estimation with the automated detection method of the initial transient period in sequential discrete-event steady-state simulation. The objective is to determine the best solution for a fully-automated simulation package which would produce a high level of credibility of the final simulation results.

One difficulty in obtaining precise estimates of performance measures for simulated systems can be the cost of computing time needed to collect the large amount of output data required. Indeed there are situations, such as estimating rare events, where, even assuming that an appropriate statistical analysis procedure is available, the cost of collecting the number of observations needed by the analysis procedure can be prohibitively large. Fortunately, the availability of inexpensive computer network resources can help computationally intensive simulations by allowing us to run parallel and distributed simulations. Therefore, where possible, we extend the previously mentioned contributions to parallel and distributed discrete-event simulation.

1.2 Sequential Steady-State Simulation

A steady-state simulation is applied for investigating the long-run behaviour of a system. Measures of performance are then steady-state parameters, characterising the steady-state distributions of output stochastic processes. There are two general procedures suggested for constructing a point estimate for the parameter of interest and a CI for that point estimate: *fixed sample size* and *sequential* for a steady-state simulation. In *fixed sample size* procedures, a single simulation run is made of a fixed number of pre-specified observations. Then a point estimate and a CI are constructed from the available data. The analyst has no control over the statistical error in this approach. Obtaining an acceptable level of statistical error is simply a matter of luck. Furthermore, no procedure in which the run-length is fixed before the simulation begins can be relied upon to produce a CI that covers the steady-state parameter with the desired probability of $1 - \alpha$ [91], [92]. *Sequential* procedures sequentially determine the length of a simulation run needed to construct an acceptable CI for the parameter [93]. With this approach, the analyst can automatically control the statistical error by specifying a stopping criterion.

The theoretical studies of sequential procedures also show that they are asymptotically consistent (as the coverage probability converges to $1 - \alpha$) and also asymptotically efficient (as the prescribed width of the CI tends to zero) for both regenerative and non-regenerative steady-state simulation [51], [149]. This asymptotic theory provides a theoretical basis for confidence in sequential procedures, regardless of any simulation output data analysis method used. Consequently, we will only examine the steady-state behaviour of systems using sequential procedures, which are very desirable in an automated simulation package.

Following Law and Kelton [93], let us consider a single run of a steady-state simulation. Firstly, let $X_1, X_2, \cdots$ be realizations of a simulation output stochastic time-stationary process X. Secondly, let $Pr(X_i \leq x|I) = F_i(x|I)$ for $i = 1, 2, \cdots$, where x is a real number and I represents the initial conditions. If $F_i(x|I) \rightarrow F(x)$ as $i \rightarrow \infty$ for all x and for any initial conditions I, then $F(x)$ is called the steady-state distribution of the output process X of interest. That is,

$$Pr(X_i \leq x|I) = F_i(x|I) \rightarrow F(x) = Pr(X \leq x) \tag{1.1}$$

as $i \rightarrow \infty$ for any initial conditions I. Therefore, $F(x)$ can be considered as a characteristic of the output process X in a steady-state when the sample size i approaches infinity.

One difficulty in estimating the steady-state parameter is that the steady-state is theoretically reachable only after an infinitely long period, but the execution of the steady-state simulation has to be completed within a finite period. This causes the distribution function $F_i(x|I)$, $1 \leq i \leq n$, of a fixed number n to be different from $F(x)$, since it will generally not be possible to choose the initial conditions I to be representative of the steady-state behaviour of the system. For example, the sample mean $\overline{X}(n) = \sum_{i=1}^{n} x_i/n$ will be a biased estimator of $\mu = E(X)$ for all finite values of n, unless observations $x_1, x_2, \cdots, x_n$ are independent and identically distributed. Various methods of approaching this problem, in the case of analysis of mean values, are discussed in [12] and [128]. Most of them (except the method of RCs) require that data collected from the initial transient period of a simulation are not used to calculate the steady-state estimates, as this can cause a significant bias in the final results of the simulation; see, for example, [167].

To eliminate the initialisation bias in the steady-state estimates, one can run the simulation experiments for a sufficiently long period to make any influence of the initial transient period negligible. However, it is difficult to ensure that the length of run chosen is long enough. On the other hand, one can collect observations only after the system has reached steady-state. However, there is also a problem in recognising whether steady-state has been reached. Determination of the length of the initial transient period can require quite elaborate statistical techniques. If a proper detection method is used, reasonable point estimates of the measures of performance needed can be established. Various detection methods have been proposed in [57], [162], and [182]. These have all been developed for the case where the steady-state mean of the system is estimated. Any method for estimating steady-state quantiles in methods other than the RCs method has not yet been developed.

When the problem of the initial transient period is solved, one is left with a stationary time series of (strongly) correlated values, and with the problem of estimating the CIs for these data. To construct the CI, various statistical techniques for obtaining accurate variances of estimators from autocorrelated simulation output data have been surveyed in [93], [128] and [135]; see Appendix B for a discussion of some of these methods. The current state-of-the-art simulation output data analysis requires extensive runs of simulation models to be made before estimates of the system's characteristics can be established. The search for robust techniques of output data analysis for a steady-state simulation continues; see, for example, [102] and [134]. In this dissertation, three selected output data analysis methods for the mean estimations and quantile estimations: non-overlapping batch means (NOBM), spectral analysis (SA), and RCs, are considered as candidates for a fully automated simulation scenario.

1.2.1 Run-Length Control in Sequential Steady-State Simulations

It is important that the run-length of the simulation be properly chosen. If the simulation is too short, the final simulation results may be highly variable.

On the other hand, if the simulation is too long, computing resources may be wasted. Sequential steady-state simulations should be run until the CI for the parameter of interest narrows to a desired width. A number of sequential run-length control methods for steady-state simulations has been proposed. Among these are sequential procedures involving: NOBM ([8], [90], [91]), SA ([63], [64], [65]), and RCs ([31], [89]). All these methods are developed for controlling the run-length by running only one simulation.

A heuristic technique, which controls the run-length by running three simulations to select the run-length of a sequential steady-state simulation is proposed in [154]. In this method, the run-length is selected by finding the point at which the three results obtained from the three independent replications are effectively the same. Sequential procedures for controlling the run-length of a simulation run, especially when several parameters are simultaneously estimated, have also been proposed in [142] for quantiles, [144] for means, and [148] for proportions.

In this section, we only discuss procedures that sequentially determine the acceptable run-length of a single simulation so that an acceptable CI with a specified statistical error for the one parameter can be constructed. Let us consider two ways of measuring the statistical error with a stopping criterion based on the half-width of the CIs for $\overline{X}(n)$ of the mean as a steady-state point estimate in a sequential steady-state simulation[1]. First, a stopping criterion can be defined as the ratio

$$\epsilon(n) = \frac{\Delta(n)}{\overline{X}(n)}, \quad 0 < \epsilon(n) < 1, \tag{1.2}$$

where $\overline{X}(n)$ is an average of collected observations $x_1, x_2, \cdots, x_n$, which are realizations of independent and identically distributed random variables $X_1, X_2, \cdots, X_n$, and

$$\Delta(n) = t_{df,1-\alpha/2}\hat{\sigma}[\overline{X}(n)], \tag{1.3}$$

is the current half-width of the CIs for the estimator at the $(1-\alpha)$ confidence level; $0 < \alpha < 1$, where $t_{df,1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of the Student t-

[1] Measuring the statistical error with a stopping criterion for the quantiles in a sequential steady-state simulation will be discussed in Chapter 5.

distribution with degrees of freedom df and $\hat{\sigma}^2[\overline{X}(n)]$ is the unbiased estimator of the variance of $\overline{X}(n)$ [128]. Depending on the output data analysis method used for estimating the variance of $\overline{X}(n)$, the degrees of freedom df is different; see Appendix B.

Equation (1.2) defines the *relative statistical error* of the CI. In a sequential simulation: if we find that $\epsilon(n) \leq \epsilon_{max}$, where ϵ_{max} is the worst acceptable relative statistical error of the final results at the $(1-\alpha)$ confidence level, $0 < \epsilon_{max} < 1$, then the simulation can be stopped at a given checkpoint[2]. Otherwise, the relative statistical error of the final results is analysed again when the next checkpoint is reached, until the final results with acceptably low statistical errors are obtained [93].

An alternative way of measuring the statistical error of the steady-state point estimate $\overline{X}(n)$ with a stopping criterion can be to apply the concept of the *absolute statistical error* $\Delta_{max} = |\overline{X}(n) - \mu|$ of the CI. In a sequential simulation, if $\Delta(n) \leq \Delta_{max}$ (where $\Delta_{max} > 0$), then the simulation can be stopped at a given checkpoint with the predefined absolute statistical error Δ_{max}. Otherwise, the sequential steady-state simulation continues until the final results with Δ_{max} are obtained [93]. This stopping criterion is very sensitive to the sample mean $\overline{X}(n)$.

Naturally, a question arises as to how well it performs in practice, in terms of producing a CI with coverage close to the desired probability $(1-\alpha)$, even though sequential procedures are intuitively appealing. The analysis of coverage is naturally limited to analytically tractable systems only, since the theoretical value of the parameter of interest has to be known. The quality of interval estimators of proportions with application to coverage analysis will be investigated in Chapter 2. In this dissertation, we only consider the stopping criterion with a relative statistical error for controlling the run-length in sequential steady-state simulations, since this is probably the most useful; see, for example, [91], [128], and [146].

[2] The point at which any new estimate is calculated is called a *checkpoint*, and the spacing between checkpoints is under the control of the analysis method. Some methods will have natural locations of checkpoints. For instances, in Batch Means a checkpoint can be located at the end of a batch or a number of batches.

One problem with such stopping criteria is that the inherently random nature of output data generated during a stochastic simulation can cause an accidental, temporal satisfaction of the stopping criterion, with the result that the final CIs of such a prematurely finished simulation run may not actually contain the exact theoretical values with the specified frequency. Rules of thumb to protect against the degradation of quality in terms of coverage of the final CIs in practical applications of fully automated sequential simulations are needed. For the coverage analysis, the stopping criterion based on the relative statistical error can include these additional conditions. Investigations of these issues will be discussed in Chapter 3.

Some commercial simulation packages offering automated control of the statistical error of the final results in a sequential steady-state simulation are Arena[3] [80], CSIM18[4], Prophesy[5], SIMPROCESS[6], Taylor II[7], and a whole family of simulation packages based on SIMSCRIPT II.5[8]. There are also packages offered as freeware for non-profit research organisations, e.g., Akaroa-2[9]. These simulation packages are compared in terms of model building, support items, and system requirements in Table 1.1. All packages have features of the automated run-length control and on-line simulation output data analysis. However, Akaroa-2 is the only one using the technique of parallel and distributed simulation to harness the computing power of a network of inexpensive workstations. CSIM18 is supported on the widest variety of platforms.

1.3 Parallel and Distributed Simulation

Simulation experiments of, for instance, computer networks, can be computationally intensive and can require long runs in order to obtain the final results at a desired level of statistical error. Research on speeding up the execution of

[3] see http://www.sm.com

[4] see http://www.mesquite.com

[5] see http://www.abstraction.com

[6] see http://www.simprocess.com

[7] see http://www.taylorii.com

[8] see http://www.caciasl.com

[9] see http://www.cosc.canterbury.ac.nz

Table 1.1: Comparisons of some simulation packages

Simulation packages	*Automated Run-Length Control*	*Graphical Modelling*	*Output Analysis Support*	*Parallel & Distributed Simulation*	*Operating Systems*
Akaroa-2	√		√	√	SunOS Solaris
Arena	√	√	√		Win 95/NT
CSIM18	√		√		Solaris Linux Windows 95/98/NT
Prophesy	√	√	√		OS/2 Win 95/98
SIMPROCESS	√	√	√		Windows 95/98/NT
SIMSCRIPT II.5	√		√		SunOS Solaris Win 95/NT
Taylor II	√	√	√		Win 95/NT

such simulations is one of the challenging issues which has attracted considerable scientific interest and effort so far; see, for example, [41], [111], [137]. The obvious solution is to speed up a simulation by executing it in a distributed way, possibly using computers linked by a local area network.

Multiprocessor and distributed systems offer high distributed processing power, many times that available with a single processor, for example in *web-based simulation.* The challenge, then, is to develop a simulation methodology that can exploit this enormous power and the economic advantage of multiprocessors and multicomputer networks to speed up simulation runs. In general, there are two classes of parallel and distributed stochastic simulation techniques: the single replication in parallel (SRIP) scenario and the multiple replications in parallel (MRIP) scenario. In this section, we discuss these two scenarios, and also load management in parallel and distributed simulations.

1.3.1 Single Replication in Parallel

Traditionally, a parallel and distributed stochastic simulation has meant running a single replication in parallel (SRIP) scenario, in which many processors cooperate in executing a single replication of a simulated process [39], [113], [153]. Research activities in the SRIP scenario have focused on developing methods for the concurrent execution of the loosely-coupled parts of large simulation models on multiprocessor computers, or multiple computers over a network. Surveys of concurrent simulation can be found, for example, in [7], [42], and [123]. Managing the execution of large partitioned simulation models efficiently with the SRIP scenario can offer reasonable speedup of a simulation, provided that a given simulation model is sufficiently decomposable. Unfortunately, this feature is not frequently observed in practice, thus this kind of a distributed simulation is strongly model-dependent [172]. Also, this scenario needs knowledge of parallel programming, which enables users to run a model simultaneously on a computer that contains two or more processors.

In the SRIP scenario, a simulation model is partitioned into several submodels, or logical processes (LPs), which are concurrently simulated by a set of processors; see Figure 1.3. The parallelism of the SRIP scenario is limited by two sequential causality constraints. Firstly, if two LPs are scheduled for the same processor, then the LP with the smaller time-stamp must be executed before the one with the larger time-stamp. Secondly, if an LP executed at a processor results in the scheduling of another LP at a different processor, then the former must be executed before the latter. The partitioning for the SRIP scenario is an important issue for minimising communication overheads between the partitions or clusters. Clustering communicating LPs together and assigning each cluster of LPs to a processor reduces the synchronisation overhead, since only inter-cluster communication requires synchronisation.

Partitioning the simulation into very fine grained objects is not an appro-

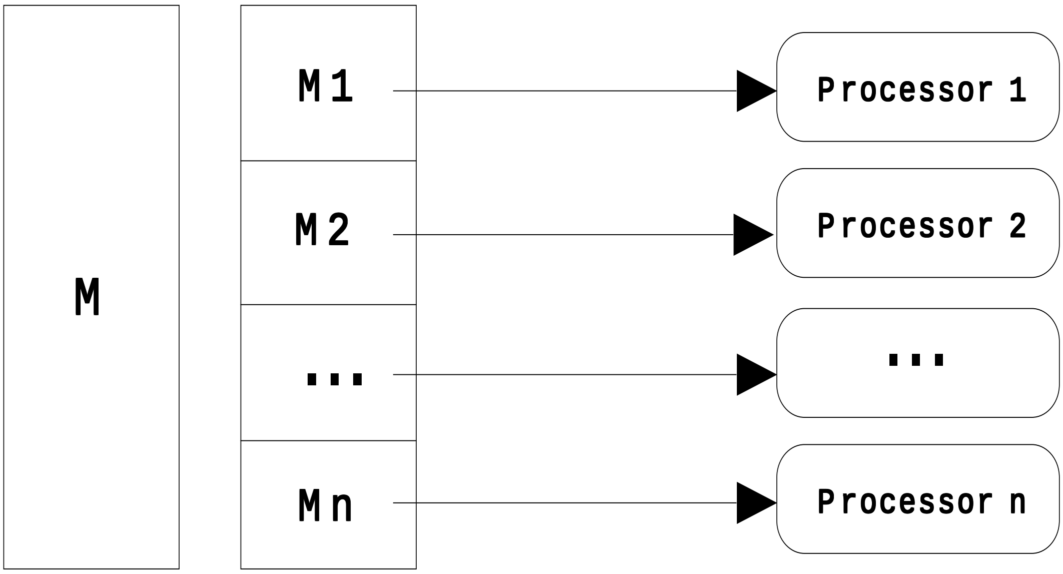


Figure 1.3: Single Replication In Parallel ($P = n$ processors, M = M1 $\cup$ M2 $\cup \cdots \cup$ Mn)

priate solution because this may lead to inappropriate computations. Optimal partitioning is a difficult problem. Many different partitioning methods have been proposed in [86]. Studies of parallel and distributed stochastic simulations often show poor performance of the system, not simply due to too many overhead messages, but because the system has an inherently low degree of concurrency, as indicated in [88].

Another important issue is synchronisation, and over the last several years research in this area has progressed along two lines: conservative and optimistic approaches. Numerous algorithms have been proposed, for example, in [42], [75], [82]. The Chandy-Misra algorithm is a well-known conservative algorithm that strictly avoids the possibility of any causality error. In this algorithm, the sequence of time-stamps on the messages sent over a link must be non-decreasing. The advantages of this method are that it avoids some of the costs associated with optimistic mechanisms, especially the state-saving overhead, and offers good potential for certain classes of problems. However, conservative algorithms appear to be poorly suited for simulating applications with poor look ahead properties, even if there is some parallelism available. As Fujimoto

[40] comments, conservative approaches cannot fully exploit the parallelism available in a given simulation application.

The Time Warp (TW) algorithm, based on the Virtual Time paradigm, is a typical optimistic algorithm: causality errors are detected, and a rollback mechanism is invoked to recover the correct state of the system. The major advantage of this method is that it offers the greatest hope as a general purpose simulation mechanism, assuming state-saving overheads can be kept to a manageable level. However, the TW algorithm has rollback and storage overhead problems since states of the processes need to be saved periodically so that they can revert to previous states when the event processing precedence is violated. Jefferson and Reiher [75] show that no conservative mechanism can beat the processing path, but at least four known optimistic mechanisms: Lazy Cancellation, Lazy Rollback, Phase Decomposition, and the Chandy-Sherman space-time family of mechanisms, are all capable of speedup. These four optimistic mechanisms are explained in detail, for example, in [40], [42], and [75].

Despite the potential speedup in the execution time due to parallel processing of subtasks on different processors, a SRIP simulation suffers from several drawbacks, in addition to the obvious overhead of distributed scheduling. One of them is the extra burden on the programmer, who must detect by himself/herself an opportunity for parallel execution, decompose the model into interacting subtasks executable in parallel, and deal with parallel coding and debugging. Furthermore, relationships between subtasks within a model may limit the degree of parallelism, especially in simulations applying the Chandy-Misra method. Hence the number of processors that may be utilised simultaneously is restricted. The resulting under-utilisation of processors can significantly reduce the expected speedup.

There are additional costs connected with the synchronisation overhead, deadlock detection and resolution, and communication between subprocesses. These phenomena also decrease the speedup by expending processor time on interprocess communication (IPC) and having idle processors whose subprocessors are blocked, waiting for input from unfinished subprocesses. Apart from consuming processor power, IPC limits the multiprocessor architectures that can be used for a SRIP simulation. For instance, in a shared memory multiprocessor, intensive IPC can create a contention in the processors-memory interconnections, causing further delays and lower speedup. Although fully connected shared memory architectures, such as crossbar or multi-stage networks, could be used to alleviate these problems, their costs grow rapidly as the number of processors increases, and they are typically limited to medium or small scale multiprocessor systems. A SRIP simulation is not fault-tolerant. If a running subtask on a processor (or a workstation in a network) fails, the simulation fails too, due to the causality between subtasks.

Load Balancing for the SRIP Scenario

Load management in parallel and distributed simulations for the SRIP scenario is very important for a judicious distribution of simulation models among processors in order to maximise the level of parallelism [6], [45], [86]. A poorly chosen load balancing technique can lead to poor performance. Load balancing techniques can be classified into static and dynamic methods.

Static load balancing techniques for the SRIP scenario may be used when processors are restricted to execute only sub-models or logical processors (LPs) that have been mapped beforehand [3]; see Figure 1.4. The advantage of this method is that the communication between LPs in a cluster can be done locally without any excessive overhead. However, the static load balancing technique cannot change the network load.

Instead of using several partitions and distributed ready queues for the LPs, dynamic load balancing techniques for the SRIP scenario can use one central scheduling queue for LPs ready to execute [25], [46]; see Figure 1.5. LPs in the centralised scheduling queue may be selected by idle processors. This can reduce the number of roll-backs and increase overall efficiency. It also avoids the problems inherent in static scheduling techniques that involve partitioning.

When the LPs do not represent the same load, or are not equally utilised, the dynamic load balancing technique performs better than the static technique, because it balances the load across the participating processors. However, choosing which implementation is best for a particular simulation model depends on the relative costs of the synchronisation and the beneficial effects of the load balancing.

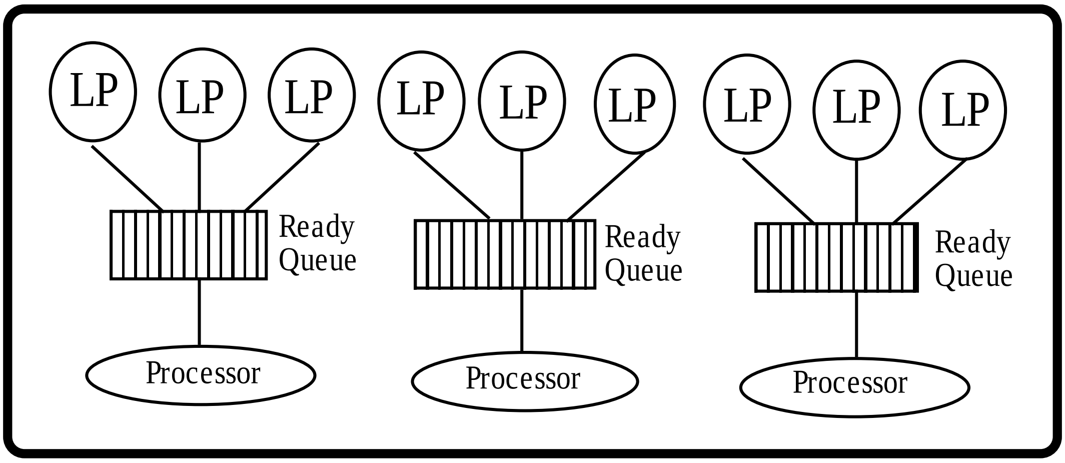


Figure 1.4: Static load balancing (taken from [3])

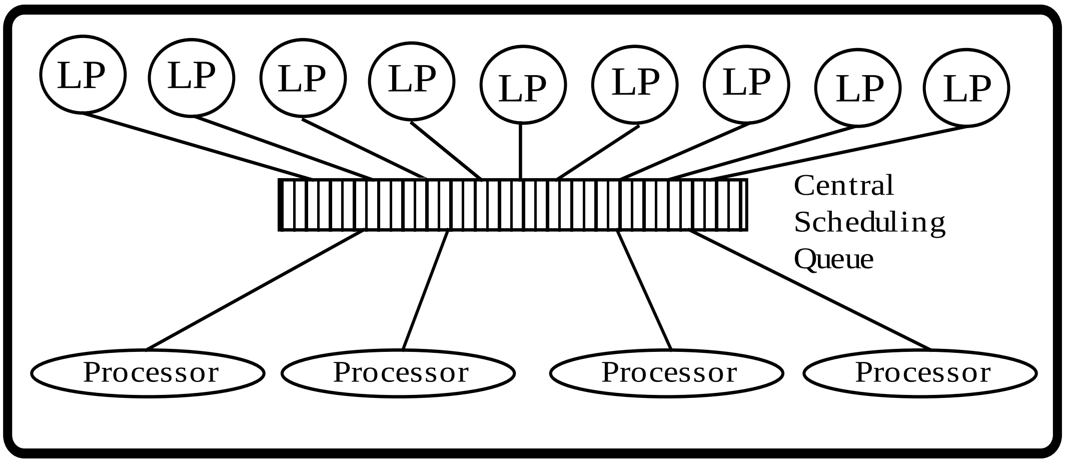


Figure 1.5: Dynamic load balancing (taken from [3])

1.3.2 Multiple Replications in Parallel

An alternative scenario is to run multiple replications in parallel (MRIP); see Figure 1.6, i.e. employing many processors, each running an independent replication of the simulated system but cooperating with central analysers that collect data to estimate the final results [136], [143], [181]. In this scenario, the entire model is replicated for execution on several processors simultaneously and the results of these replications are then averaged. Therefore, collecting a sufficient amount of simulation output data for sequential analysis can be sped up if the output data are produced in parallel by multiple simulation engines running statistically identical simulation processes.

User-friendly simulation packages for running a parallel and distributed stochastic simulation based on the MRIP scenario, such as (i) Akaroa-2 ([28], [181]) at the University of Canterbury, Christchurch, New Zealand, (ii) EcliPse ([151]) at Purdue University, West Lafayette, USA, and (iii) QNSim ([147]) at the University of Helsinki, Finland, have been developed to fully use the enormous distributed power of modern computer networks.

The MRIP scenario is a conceptually simple scenario and can potentially

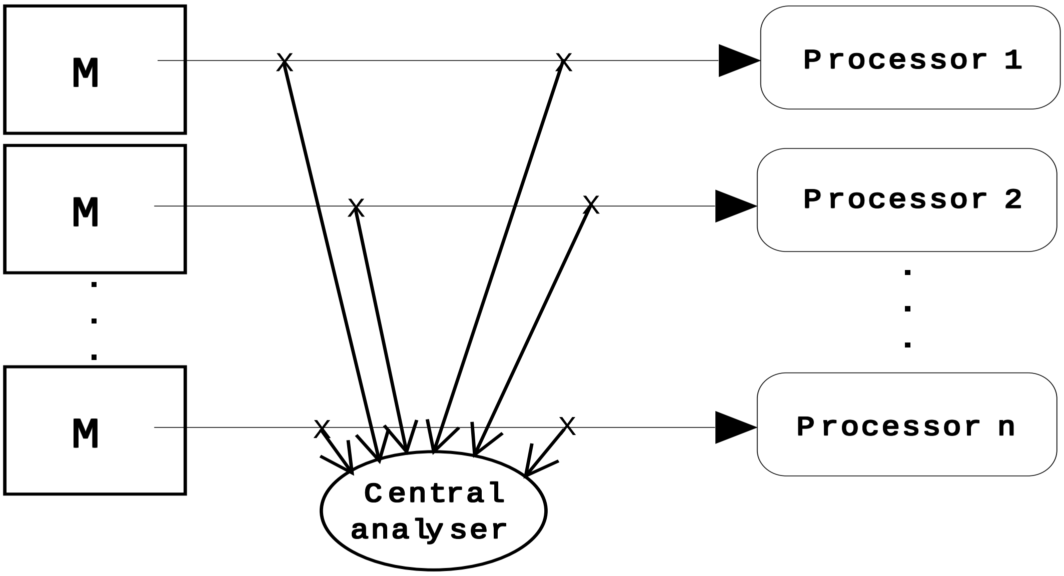


Figure 1.6: Multiple Replications In Parallel ($P = n$ processors)

be applied to any model. Furthermore, the MRIP scenario is suitable for execution on multiprocessors as well as multicomputer networks. Heidelberger [60] shows that the MRIP scenario produces statistically better results than the SRIP scenario, if the effect of the initialisation bias is sufficiently covered (or negligible), and when the memory available to each processor (in the case of parallel execution on a multiprocessor) or each participating computer (in the case of execution on a network of computers) is not a limiting factor, i.e. when the physical user memory available to each processor is sufficient to store the working set of the replication.

P independent replications of a simulation from P independent processors are launched when the simulation begins; each replication is run in a parallel time stream to the others. When the number of observations for the estimate of a parameter reaches a checkpoint for that parameter, a local point and interval estimate of that parameter is produced, and then the estimates are sent to the global control process, responsible for estimating that parameter and for checking out sequentially the stopping criterion to stop the run. No coordination is required among processors. The MRIP scenario is the most effective if it is applied in homogeneous networks.

It is possible that when the MRIP scenario is applied in a heterogeneous network, with one processor much faster than others, slower processors may not be able to contribute to the parallel production of data, since none of them would reach its first checkpoint when the fastest processor stops the whole simulation by generating the required number of observations. In [112], it is suggested that limitations on the computing resources in the MRIP scenario are necessary for two reasons. Firstly, executing a replication on a slow computer may significantly increase the workload, which affects the performance of other applications on that computer. Secondly, adding an extra slow computer may increase the time complexity of the MRIP scenario.

When using the MRIP scenario on a large number of processors, one expects to get highly accurate estimates after only a relatively short time. Potentially it can offer speedup with the number of processors involved, while the speedup under the SRIP scenario depends very much on the partitioning and load balancing techniques. At first glance, the MRIP scenario produces P-fold speedup, (i.e., a reduction in completion time) over a sequential (one processor) simulation having the same variance (P is the number of processors involved). A MRIP simulation requires that all processors have enough memory to contain the entire simulation program, so it may not be practical in some cases. One advantage of multiprocessor simulations is to permit, based on time and memory constraints, much larger and more realistic simulations than has been possible on a single processor.

However, the MRIP scenario is inappropriate in the following cases. First is the case of a single replication that cannot be fitted within a memory of a single processor. This may be due to an exceptionally large model [143]. Secondly, if the variance of the estimated values of interest is very small, the output is nearly deterministic and a large number of replications is merely a waste of computing resources.

Load Balancing for the MRIP Scenario

Load balancing techniques, based on static and dynamic methods, for the MRIP scenario have been proposed in [112]. However, in a steady-state simulation under this scenario, the load balancing techniques are not really required.

1.4 Organisation of the Thesis

In many simulation studies, the analyst is interested not only in the point and interval estimates of mean values, but also in other characteristics such as variances, quantiles and proportions (or probabilities) of the simulation output. The quality of all these characteristics has to be investigated. Generally, this can be done by coverage analysis. This is one of the applications in confidence interval (CI) estimations of proportions. Therefore, we investigate CI estimators for proportions in a sequential steady-state simulation for a fully automated simulation package in Chapter 2. Three interval estimators, based on the normal distribution, the *arcsin* transformation, and the F distribution, are studied in sequential steady-state simulations of the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems. The most reliable interval estimator for proportions found in Chapter 2 will be used to assess the quality of simulation output data analysis methods in Chapter 3 - Chapter 5. Of course, it can also be used for estimating proportions or probabilities in practice.

Chapter 3 discusses a problem associated with the fact that a stochastic simulation can be stopped accidentally when the stopping criterion is only temporarily observed in the case of mean value estimations. To eliminate this problem, we propose solutions that can substantially increase the reliability of results in a fully automated simulation package. The results of the performance evaluation of the proposed heuristic rules obtained using the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems are presented. In Chapter 4, we investigate the method of RCs for simulation output data analysis for a fully automated sequential steady-state simulation, along with two other methods: NOBM and SA, in the case of mean value estimations. In particular, we study a problem of the sequential method of RCs and propose a possible solution to eliminate it. The results of the performance evaluation with and without the proposed solution, in terms of coverage analysis using the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems, are also presented.

Quantiles are often used to give a more complete description of the distribution, since the mean value of a random variable is seldom sufficient as summary of an entire distribution. However, traditional quantile estimation (QE) has its own limitations: computation time for sorting the entire sequence, and memory for storing the entire sequence. To overcome these limitations, several approaches have been proposed, but most approaches for a fixed sample size simulation. In Chapter 5, QE in sequential steady-state simulation based on three methods: *linear* and *batching* QE for the method of RCs, and *spectral* P^2 QE for the method of non-RCs, is investigated to discover the best method for a fully automated simulation tool. The numerical results of the coverage analysis of these estimators are presented. Methods of sequentially detecting the initial transient period for QE are also investigated in Chapter 5.

A problem of sequential steady-state simulation is that sound simulation studies require very long runs to obtain the final results with acceptable accuracy. The obvious solution is to speed up these simulations by executing them on a multiprocessor or distributed computer system. Therefore, the speedup should be achieved when estimating mean values or quantiles on a multiprocessor or distributed computer system using any methods of simulation output data analysis. To have more conviction on the sequential estimation methods of means and quantiles to be implemented in a fully automated simulation package like Akaroa-2, which uses techniques of the MRIP scenario for speedup a simulation, we have investigated them in terms of speedup in Chapter 6. Theoretical limitations on the speedup of sequential stochastic simulations under the MRIP scenario, based on [133], are discussed. We also present the empirically obtained speedup for this scenario when estimating mean values and quantiles for the different methods of simulation output data analysis.

Chapter 7 summarises the main contributions of this thesis, in particular for a fully automated simulation tool, in both distributed and non-distributed stochastic simulations, and also recommends further research.

Chapter 2

COVERAGE AS THE PERFORMANCE MEASURE OF SEQUENTIAL STEADY-STATE SIMULATION

2.1 Introduction

In many simulation studies of computer systems and telecommunication networks, the analyst is interested not only in the point and interval estimates of mean values of waiting times and delays, but also in other characteristics such as variances, quantiles and proportions (or probabilities) of the simulation output. Following the most basic principles of scientific experimentation, the final result from performance evaluation studies of stochastic dynamic systems, by means of discrete-event simulation, should always be presented with some estimate of their statistical errors. These errors are usually measured by the half-width of the final CIs. However, the methods proposed for estimating the CIs of different performance measures (such as mean values, variances, probabilities, quantiles, etc.) are based on different approximations, which cause the experimental confidence level (or coverage) of the final CIs to differ significantly from the assumed (theoretical) confidence level.

There are some theoretical studies of a coverage error for CIs arising in simulation output data analysis (see [47]), a coverage function (which is defined for all confidence levels between zero and one) to measure the robustness of CIs (see [159]), and coverage properties of CIs based on the Bayesian *posterior* probability (see [155]). However, experimental analysis of coverage is still required to assess the quality of practical implementations of the methods used for determining the final CIs, especially in the context of stochastic steady-state simulation. The aim of experimental coverage analysis is to find the best method(s) (in the sense of coverage) that could be applied in simulation output data analysis.

Statistical analysis of the output data of stochastic steady-state simulation is made difficult by the degree of serial correlation often present. Various methods such as batch means, SA, RCs, etc are used to overcome this difficulty. One of the important measurements of the robustness of any simulation output analysis method is the coverage, defined as the proportion of such CIs that contain the true value of the parameter, obtained from a number of independent replications. The estimate obtained using any good method of analysing simulation output data should have narrow and stable CIs, and at the same time the probability of such an interval containing the true value of the estimated performance measure should be very close to the assumed confidence level.

As an example, Figure 2.1[1] shows typical results of such a coverage analysis, where the method of non-overlapping batch means (NOBM) has been used to analyse the mean waiting time of an $M/M/1/\infty$ queueing model in a sequential steady-state simulation. The actual coverage of the CIs drops away from the assumed confidence level (95%) as the traffic intensity increases. This

[1] Each replication for coverage analysis was obtained with the required statistical error of 10% or better, and sequential coverage analysis was undertaken assuming that the required statistical error of the final result was 5% or better, both at a confidence level of 0.95.

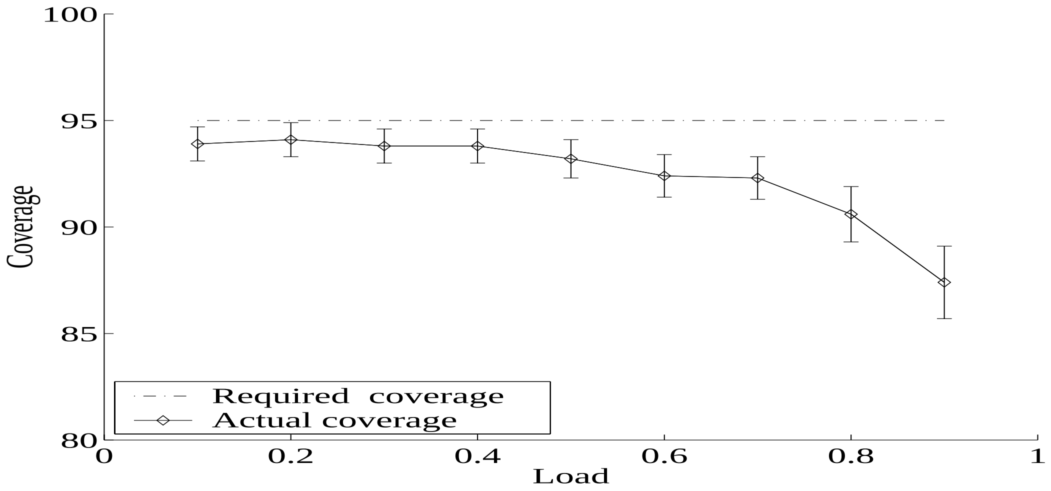


Figure 2.1: Coverage analysis of the method of non-overlapping batch means (simulation of the $M/M/1/\infty$ queueing system)

may be related to the autocorrelations of waiting times[2] increasing rapidly as the full traffic load of $\rho = 1.0$ is approached as shown by Daley in [23]. As discussed in Appendix B.1, it may also be affected by the difficulty in choosing the optimal batch size for reducing or eliminating autocorrelations, especially in the case where very strong autocorrelations exist between collected observations. Therefore, in such a case one needs to collect a huge number of observations in order to have credible final simulation results with the required statistical error. As discussed in [19], a larger batch size is needed to obtain approximately uncorrelated batch means if observations are more correlated. In Figure 2.1, the major reason for poor coverage in heavier traffic intensities may be that, even with a sophisticated automatic algorithm for batch size selection, autocorrelations between batch means in heavily loaded traffic may not be eliminated.

There are a number of factors to consider when analysing coverage experimentally. First, analysis is limited to analytically tractable systems, since the theoretical value of the parameter of interest has to be known [134]. Because

[2] In general, if we increase the service times or decrease the interarrival times in concerning queueing systems, then the system becomes more congested and hence the waiting times of successive customers become more correlated [9].

of that, it has been claimed that there is no justification for experimental coverage analysis, since there is no theoretical basis for extrapolating the results obtained for simple, analytically tractable systems to more complex systems, which are the subjects of practical simulation studies [37]. On the other hand, no theory of coverage for finite sample sizes exists, and, in this situation, experimental coverage analysis of analytically tractable systems remains the only method available for testing the validity of methods proposed for simulation output analysis. Certainly nobody should be ready to accept a method of simulation output data analysis showing very poor quality in experimental studies of coverage.

Coverage analysis also requires the execution of multiple, independent replications of simulations. Very large numbers of replications are often needed to determine the coverage with satisfactory precision. For example, typical experiments on the waiting times in an $M/M/1/\infty$ queueing system, with a traffic intensity of 0.9 as in Figure 2.1, require about 1,900 independent replications, where each replication measures the waiting times of about 100,000 customers to ensure having the required statistical error of the final result. This indicates that the coverage study should be analysed on the basis of a large number of replications. However, many coverage studies appear to have used for too few replications of between 10 to 200: see, for example, [2], [62], [65], [77], [79], [91], [139], [158], [161], [164], [168], and [178]. We have found one study which used 500 replications [93], and four studies, in [56], [68], [78], and [148], which used 1000 replications. In these cases, the estimates of coverage can be questioned, since they may be obtained from CIs which do not cover the true value of the parameter of interest.

As argued in Chapter 1, in general, sequential analysis of simulation output data is accepted as the most efficient way of securing a certain level of accuracy in the final results. For this reason, as also argued in [134], coverage analysis should be performed sequentially to ensure that results are statistically acceptable. As in the case of ordinary sequential simulation, sequential coverage analysis is continued until the final result is obtained with sufficient precision.

Sequential analysis, however, raises additional problems. A major problem[3] is that some of the simulation experiments may stop after an abnormally short run, when the stopping criterion for the sequential simulation is temporarily satisfied.

The quality of any method used in sequential simulations can be measured by coverage analysis. Abnormally short runs produced in sequential simulations can obviously affect the quality measured by coverage analysis. Therefore, interval estimators of the proportions used for determining the precision of coverage play a crucial role in its sequential analysis, since abnormally short runs can be excluded from the final result by the sequential coverage analysis. The conventional interval estimator based on the normal approximation has been widely used in coverage analysis (see for example [77], [93], [116], [120], [134]). Alternative interval estimators of proportions are discussed in [10] and [59]. Recently, one of these estimators (based on the *arcsin* transformation) has been used for the analysis of proportions in sequential steady-state simulations [139], [145], [148]. However, as yet a comparative study of these three estimators has not been undertaken.

In this chapter we document our search for the best interval estimator of proportions, which could be applied to coverage analysis in a fully automated sequential steady-state simulation. Three interval estimators, based on the normal distribution, the *arcsin* transformation, and the F distribution, as described in Section 2.2, are compared. The results of the performance evaluation of these estimators are presented in Section 2.3. Comparisons of the three interval estimators with exact values, calculated using the binomial distribution, are also presented in Section 2.4. Taking account of these results, some rules for the sequential analysis of coverage have been summarised in Section 2.5. Experimental results of coverage analysis in the sequential steady-state simulations of the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems, applying these proposed rules, are also presented in Section 2.5. Conclusions can be found in Section 2.6.

[3] This problem will be investigated in detail in Chapter 3.

2.2 Interval Estimators for Coverage Analysis

To estimate coverage, we need point and interval estimates of the proportion of sample CIs which contain the true value of the parameter of interest. If each of the experiments executed for coverage analysis is statistically independent from others, then an exact CI for the estimated proportion is obtained using the binomial distribution [101].

A binomial experiment consists of repeated trials, each with two possible outcomes, which may be labelled *success* or *failure*. The point estimator of the proportion p in a binomial experiment is simply given by the statistic

$$\hat{p} = \frac{count\ of\ successes\ in\ sample}{size\ of\ sample} = \frac{X}{n}. \tag{2.1}$$

If a binomial experiment can result in a success with probability p and a failure with probability $(1-p)$, then the probability distribution of the binomial random variable X, the number of successes in n independent experiments, is

$$b(x;n,p) = \left(\begin{array}{c} n \\ x \end{array} \right) p^x (1-p)^{n-x}, \ \ x = 0, 1, \ldots, n. \tag{2.2}$$

The accuracy with which $\hat{p}$ estimates an unknown proportion p can be assessed by the width of its CI at a given confidence level, i.e, by the probability

$$Pr(\hat{p} - \Delta_1 \leq p \leq \hat{p} + \Delta_2) = 1 - \alpha, \tag{2.3}$$

where $\hat{p}$ is the estimate of the proportion p, Δ_1 and Δ_2 are the offset for the lower and upper limit of the CIs of p, and $(1-\alpha)$ is the confidence level, $0 < \alpha < 1$. Ideally, this would mean that if the simulation experiment is repeated sufficiently many times, the resulting CI would contain the parameter p in $100(1-\alpha)\%$ of cases [134].

As discussed in Section 2.1, the robustness of any methods of data collection and analysis is usually measured in the context of the coverage of CIs. Sauer [156] proposed that the method used for determining the CI of the point estimate at a given confidence level $(1-\alpha_0)$ is considered as valid if the upper limit of the CIs of p equals at least $(1-\alpha_0)$. The coverage is defined as the frequency with which CIs $(\hat{p}-\Delta_1, \hat{p}+\Delta_2)$ contain the true parameter (i.e., the theoretical value) at a given confidence level $(1-\alpha)$; see Figure 2.2. In the example of Figure 2.2, the coverage is 80% since 8 out of 10 CIs contain the theoretical value.

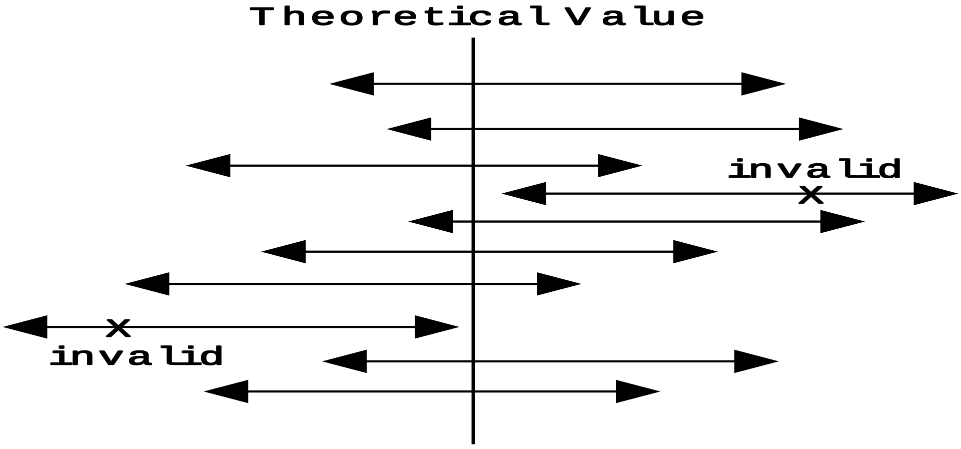


Figure 2.2: Valid and invalid CIs in coverage analysis

To determine Δ_1 and Δ_2 in the CIs $(\hat{p}-\Delta_1, \hat{p}+\Delta_2)$, we need the exact distribution of $\hat{p}$, or at least to know $Var(\hat{p})$. Calculating exact confidence limit values of p is possible only using Equation (2.2). However, expanding and inverting the polynomials of the order n becomes impractical, even using a computer algebra system, as n increases. The time complexity of the polynomials of the order n is $O(p^n)$ [67]. Therefore, some approximation methods for a binomial distribution have been suggested. Three interval estimators of the proportion p, based on the normal distribution, the *arcsin* transformation, and the F distribution are described in the following subsections. Detailed procedures for these are discussed in [99].

2.2.1 Interval Estimator Based on the Normal Distribution

The interval estimator based on the normal distribution for finding a CI for the binomial parameter p, $0\leq p \leq 1$, approximates the binomial distribution of

$\hat{P}$ by the normal distribution, with mean $\hat{p}$ and variance $\hat{p}(1-\hat{p})/n$ [59].

For large n, the random variable

$$Z = \frac{\hat{P} - \hat{p}}{\sqrt{\hat{p}(1-\hat{p})/n}} \tag{2.4}$$

is approximately standard normal; see, for example, [173]. Thus, an approximate CI for the proportion p is

$$Pr(\hat{p} - z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} < p < \hat{p} + z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}) \approxeq 1 - \alpha. \tag{2.5}$$

Note that this is a symmetric CI.

The accuracy of the normal approximation improves as the sample size n increases. However, it is most accurate when p is close to 1/2, and becomes quite inaccurate when p is near 0 or 1, mostly due to the skewed nature of the binomial distribution. Unfortunately, this is exactly the situation in simulation coverage analysis, where typically p is between 0.9 and 0.99. Thus, we need an interval estimator for coverage analysis which can produce an asymmetric CI in this region.

2.2.2 Interval Estimator Based on the *Arcsin* Transformation

An asymmetric CI for proportions based on the *arcsin* transformation was originally proposed by R. A. Fisher (see [59] for detailed discussion). On the basis of the relationship between the mean p and variance $p(1-p)/n$ for the proportion $\hat{p} = X/n$, one can determine a function $\hat{Y} = g(\hat{p})$ in such a manner that the variance of the transformed variable $\hat{Y}$ is independent of p. This leads to the transformation function $\hat{Y} = 2\, arcsin\sqrt{\hat{p}}$ with variance $\sigma^2(\hat{Y}) = 1/n$ [59].

An approximate 100(1 - α)% CI for a proportion using this transformation is constructed by ($\hat{p}_l$, $\hat{p}_u$), where

$$\hat{p}_l = sin(l/2)^2$$

and

$$\hat{p}_u = sin(u/2)^2. \tag{2.6}$$

Here

$$l = \ arcsin\sqrt{\hat{p} - 1/(2n)} - z_{1-\alpha/2}/\sqrt{n}$$

and

$$u = \ arcsin\sqrt{\hat{p} + 1/(2n)} + z_{1-\alpha/2}/\sqrt{n}. \tag{2.7}$$

In these formulae, $\hat{p}$ is the sample proportion, $z_{1-\alpha/2}$ is the $(1 - \alpha/2)$ quantile of the standard normal distribution, and n is the sample size [59], [145], [148].

2.2.3 Interval Estimator Based on the F Distribution

CIs for proportions can also be formulated from the relationship of the F and binomial distributions. The ratio of two successive terms in a binomial distribution $(x; n, p)$ is

$$\frac{n-x}{x+1}\frac{p}{1-p}, \qquad x = 0, 1, \ldots, n-1, \tag{2.8}$$

where x is the observed number of successes in the sample; see Equation (2.1). Using the transformations shown, for example, in [1] and [59], the quantiles of the binomial distribution can be obtained from those of the F distribution, as

$$Pr\{F(df_1, df_2) < \frac{n - n\hat{p}}{n\hat{p}+1}\frac{p}{1-p}\} =$$

$$Pr\{\frac{(n\hat{p}+1)F(df_1, df_2)}{(n - n\hat{p}) + (n\hat{p}+1)F(df_1, df_2)} < p\}, \tag{2.9}$$

where $F(df_1, df_2)$ is a random variable with the F distribution of $df_1 = 2 * (n\hat{p} + 1)$ and $df_2 = 2 * (n - n\hat{p})$ degrees of freedom.

Thus, a 100(1 - α)% CI for a proportion is given by $(\hat{p}_l, \hat{p}_u)$, where

$$\hat{p}_u = \frac{(n\hat{p}+1)f_{1-\alpha/2}(df_1, df_2)}{(n - n\hat{p}) + (n\hat{p}+1)f_{1-\alpha/2}(df_1, df_2)}$$

and

$$\hat{p}_l = \frac{n\hat{p}}{n\hat{p} + (n - n\hat{p} + 1) f_{1-\alpha/2}(df_3, df_4)}. \tag{2.10}$$

Here, n is the sample size, and $f_{1-\alpha/2}(df_1, df_2)$ is the $(1-\alpha/2)$ quantile of the F distribution with (df_1, df_2) degrees of freedom, where $df_1 = 2 * (n\hat{p} + 1)$ and $df_2 = 2 * (n - n\hat{p})$, while $f_{1-\alpha/2}(df_3, df_4)$ is the $(1-\alpha/2)$ quantile of the F distribution with (df_3, df_4) degrees of freedom, where $df_3 = 2 * (n - n\hat{p} + 1)$ and $df_4 = 2 * n\hat{p}$ [59].

2.3 Performance Evaluation of Three Interval Estimators

To find the most reliable interval estimator for coverage analysis, we investigated the properties of three interval estimators of proportions, based on the normal distribution, the *arcsin* transformation, and the F distribution. The quality of these three interval estimators was evaluated by applying them in sequential coverage analysis of the sequential estimation of steady-state means.

To show the performance of the three interval estimators, the SA/HW method (spectral analysis in its version proposed by Heidelberger and Welch [63], see Appendix B.2 for a detailed discussion) was considered as it has proved to be quite a satisfactory method for the estimation of CIs in sequential parallel simulation of steady-state means [134]. The results reported in this section were obtained during the performance evaluation of the SA/HW method for the coverage analysis of the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$[4] queueing systems when estimating the mean response times and stopping the sim-

[4] H_2 means the hyperexponential distribution of degree 2 that can be represented as the two exponential distribution in parallel. The parameters that need to be specified for the $M/H_2/1/\infty$ queueing system are the mean customer arrival rate, the mean service time per customer, and the squared coefficient of variation for service time (C^2). Then, the probability of selecting each exponential being α_1 and α_2, and the mean values of the exponential being μ_1 and μ_2 are calculated with C^2. A convenient method of doing this is suggested in [4]. As assuming $C^2 = 5$, we have obtained $\alpha_1 = 0.09175$, $\alpha_2 = 1$ - α_1, $\mu_1 = 0.18350$, and $\mu_2 = 1.81650$ by applying the algorithm in [4].

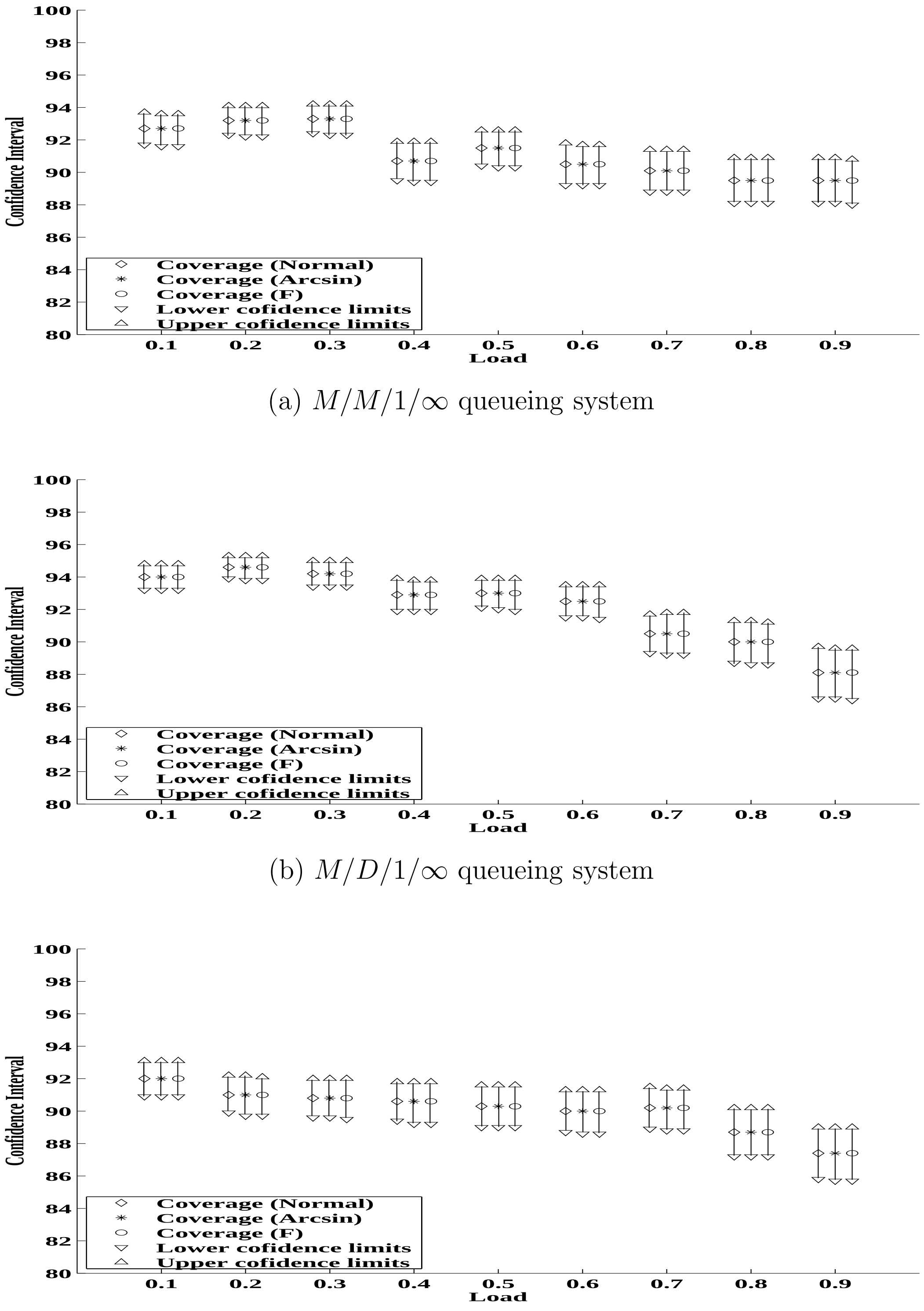
ulation experiments when the final steady-state results reached the required relative statistical error of 5% or less, at the 0.95 confidence level.

All coverage results were filtered for unusually short simulation runs, (since they produce unrepresentative results), by discarding runs shorter than a threshold (one standard deviation below the mean of the run-lengths) [134]. These steps should ensure that the results come from a well-managed simulation experiment. Furthermore, at least 200 CIs not covering the theoretical value were collected. This number of observed 'invalid' CIs was recommended in [134], to ensure that the coverage estimates are obtained from representative samples.

The simulations were executed using the Akaroa-2 simulation package [28], a controller of sequential stochastic discrete-event simulation. Properties of the SA/HW method were investigated in sequential stochastic simulations. The results for each interval estimator in simulations of the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems in terms of CIs of the coverage are presented in Figure 2.3 and Table 2.1. It can be seen that the CIs at different traffic intensities using the normal distribution, the *arcsin* transformation, and the F distribution[5] are quite similar.

However, one can see that the CIs obtained using the interval estimator based on the normal distribution are always symmetric. This means that such a symmetric CI can be invalid, since it can have its lower limit less than zero or its upper limit greater than one. This cannot happen with the other estimators, producing the asymmetric CIs, including their applications in estimation of very lower or very higher proportions.

[5] The numerical values for the F distribution were obtained from a carefully validated implementation of the method proposed in [1] and [138].



(a) $M/M/1/\infty$ queueing system

(b) $M/D/1/\infty$ queueing system

(c) $M/H_2/1/\infty$ queueing system

Figure 2.3: CIs of coverage of SA/HW using the normal distribution, the *arcsin* transformation, and the F distribution

Table 2.1: Coverage and its CIs of SA/HW using the normal distribution, the *Arcsin* transformation, and the F distribution (when estimating the mean response time at a confidence level = 0.95 with a statistical error $\leq 5\%$)

(a) $M/M/1/\infty$ queueing system

ρ	*Normal*		*Arcsin*		F	
	Coverage	**CIs**	**Coverage**	**CIs**	**Coverage**	**CIs**
0.1	92.7	91.7, 93.7	92.7	91.6, 93.6	92.7	91.6, 93.6
0.2	93.2	92.3, 94.1	93.2	92.2, 94.1	93.2	92.2, 94.1
0.3	93.3	92.4, 94.2	93.3	92.3, 94.2	93.3	92.3, 94.2
0.4	90.7	89.5, 91.9	90.7	89.4, 91.9	90.7	89.4, 91.9
0.5	91.5	90.4, 92.6	91.5	90.3, 92.6	91.5	90.3, 92.6
0.6	90.5	89.2, 91.8	90.5	89.2, 91.7	90.5	89.2, 91.7
0.7	90.1	88.8, 91.4	90.1	88.8, 91.4	90.1	88.8, 91.4
0.8	89.5	88.1, 90.9	89.5	88.1, 90.9	89.5	88.1, 90.9
0.9	89.5	88.1, 90.9	89.5	88.1, 90.9	89.5	88.0, 90.8

(b) $M/D/1/\infty$ queueing system

ρ	*Normal*		*Arcsin*		F	
	Coverage	**CIs**	**Coverage**	**CIs**	**Coverage**	**CIs**
0.1	94.0	93.2, 94.8	94.0	93.2, 94.8	94.0	93.2, 94.8
0.2	94.6	93.9, 95.3	94.6	93.8, 95.3	94.6	93.8, 95.3
0.3	94.2	93.4, 95.0	94.2	93.4, 95.0	94.2	93.4, 95.0
0.4	92.9	91.9, 93.9	92.9	91.9, 93.8	92.9	91.9, 93.8
0.5	93.0	92.1, 93.9	93.0	92.0, 93.9	93.0	91.9, 93.9
0.6	92.5	91.5, 93.5	92.5	91.5, 93.5	92.5	91.4, 93.5
0.7	90.5	89.3, 91.7	90.5	89.2, 91.8	90.5	89.2, 91.8
0.8	90.0	88.7, 91.3	90.0	88.6, 91.3	90.0	88.6, 91.2
0.9	88.1	86.5, 89.7	88.1	86.5, 89.6	88.1	86.4, 89.6

(c) $M/H_2/1/\infty$ queueing system

ρ	*Normal*		*Arcsin*		F	
	Coverage	**CIs**	**Coverage**	**CIs**	**Coverage**	**CIs**
0.1	92.0	90.9, 93.1	92.0	90.9, 93.1	92.0	90.9, 93.1
0.2	91.0	89.9, 92.2	91.0	89.7, 92.2	91.0	89.7, 92.1
0.3	90.8	89.6, 92.0	90.8	89.6, 92.0	90.8	89.5, 92.0
0.4	90.6	89.4, 91.8	90.6	89.2, 91.8	90.6	89.2, 91.8
0.5	90.3	89.0, 91.6	90.3	89.0, 91.6	90.3	89.0, 91.6
0.6	90.0	88.7, 91.3	90.0	88.6, 91.3	90.0	88.6, 91.3
0.7	90.2	88.9, 91.5	90.2	88.8, 91.4	90.2	88.8, 91.4
0.8	88.7	87.2, 90.2	88.7	87.2, 90.2	88.7	87.2, 90.2
0.9	87.4	85.8, 89.0	87.4	85.7, 89.0	87.4	85.7, 89.0

2.4 Comparisons of Three Interval Estimators with Exact Values

Taking a closer look at the three interval estimators, the CIs of proportions using the normal distribution, the *arcsin* transformation, and the F distribution at a given confidence level (1 - $\alpha = 0.99$) and a sample size[6] $n = 20$ are depicted in Figure 2.4. The upper limits of the CIs of proportions using each interval estimator and the 'exact' upper limits of the CIs of proportions which are calculated by the binomial probability function are in Table 2.2. The rela-

[6] We have chosen a small sample size of twenty for visibility in the figure and to obtain the exact values of proportions from the binomial distribution (as discussed in Section 2.2, it is impossible to calculate them at a large sample size). The similar results obtained from larger sample sizes will be presented later.

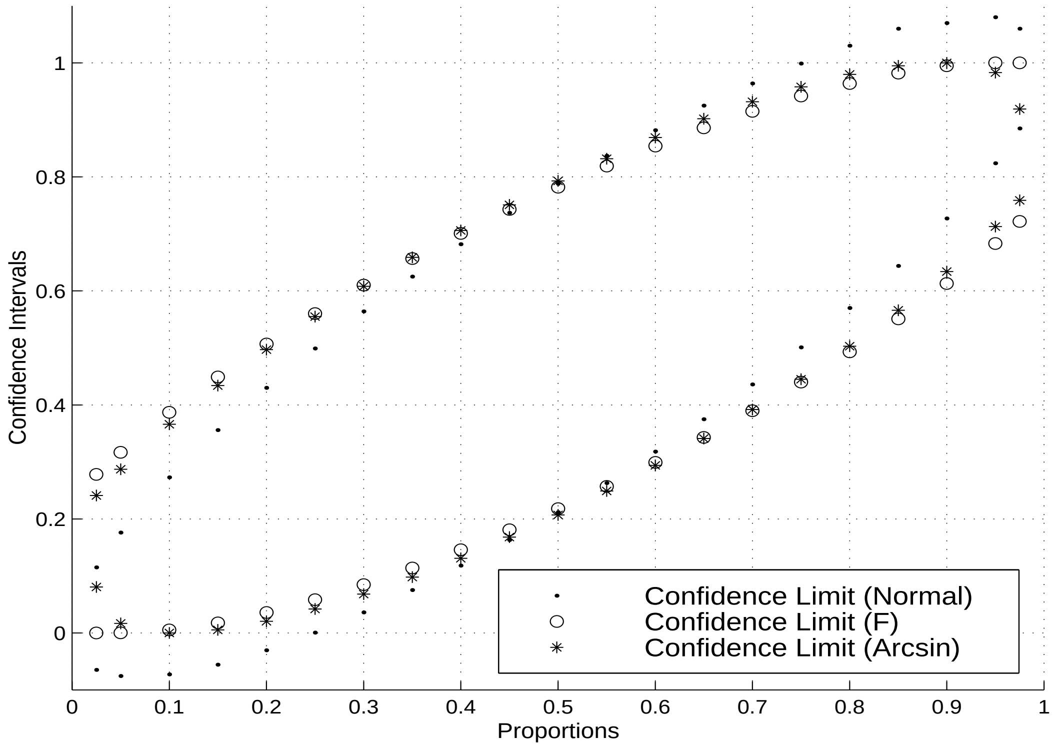


Figure 2.4: The CIs of proportions using the normal distribution, the *arcsin* transformation, and the F distribution ($\alpha = 0.01$ & $n = 20$)

Table 2.2: Upper limits of CIs of proportions ($\alpha = 0.01$ & $n = 20$)

Proportions	*Exact Values*	*Normal Dist.*	*Arcsin Transf.*	*F Distribution*
0.0	0.206	0.0	0.187	0.233
0.05	0.289	0.176	0.287	0.317
0.1	0.358	0.273	0.366	0.387
0.15	0.421	0.356	0.434	0.449
0.2	0.478	0.430	0.497	0.507
0.25	0.532	0.499	0.555	0.560
0.3	0.583	0.564	0.608	0.610
0.35	0.631	0.625	0.659	0.657
0.4	0.677	0.682	0.706	0.701
0.45	0.720	0.737	0.751	0.743
0.5	0.761	0.788	0.793	0.782
0.55	0.800	0.837	0.832	0.819
0.6	0.837	0.882	0.869	0.854
0.65	0.871	0.925	0.902	0.886
0.7	0.902	0.964	0.932	0.915
0.75	0.931	0.999	0.958	0.942
0.8	0.956	1.03	0.980	0.964
0.85	0.977	1.06	0.995	0.982
0.9	0.992	1.07	1.0	0.995
0.95	0.999	1.08	0.983	1.0
1.0	1.0	1.0	0.940	1.0

tive inaccuracy of the upper confidence limits of the three interval estimators when compared to the 'exact' values of the upper confidence limit are in Table 2.3. For higher proportions, the interval estimator based on the F distribution produces the closest values to the exact values, while the interval estimator based on the normal distribution produces values exceeding the upper limit of 1.0.

Table 2.3: Relative inaccuracy of upper confidence limits of proportions ($\alpha = 0.01$ & $n = 20$)

Proportions	*Normal Distribution*	*Arcsin Transformation*	*F Distribution*
0.0	- 100 %	- 9.2 %	+ 13.1 %
0.05	- 39.1 %	- 0.7 %	+ 9.7 %
0.1	- 23.7 %	+ 2.2 %	+ 8.1 %
0.15	- 15.4 %	+ 3.1 %	+ 6.7 %
0.2	- 10.0 %	+ 4.0 %	+ 6.1 %
0.25	- 6.2 %	+ 4.3 %	+ 5.3 %
0.3	- 3.3 %	+ 4.3 %	+ 4.6 %
0.35	- 1.0 %	+ 4.4 %	+ 4.1 %
0.4	+ 0.7 %	+ 4.3 %	+ 3.5 %
0.45	+ 2.4 %	+ 4.3 %	+ 3.2 %
0.5	+ 3.5 %	+ 4.2 %	+ 2.8 %
0.55	+ 4.6 %	+ 4.0 %	+ 2.4 %
0.6	+ 5.4 %	+ 3.8 %	+ 2.0 %
0.65	+ 6.2 %	+ 3.6 %	+ 1.7 %
0.7	+ 6.9 %	+ 3.3 %	+ 1.4 %
0.75	+ 7.3 %	+ 2.9 %	+ 1.2 %
0.8	+ 7.7 %	+ 2.5 %	+ 0.8 %
0.85	+ 8.5 %	+ 1.8 %	+ 0.5 %
0.9	+ 7.9 %	+ 0.8 %	+ 0.3 %
0.95	+ 8.1 %	- 1.6 %	+ 0.1 %
1	0 %	- 6.0 %	0 %

To see whether the interval estimator based on the normal distribution produces invalid CIs when the number of sample size is increased, we tested it for larger sample sizes n (ranging between 10,000 and 1,000,000) and the α of 0.05 - 0.001. The results are presented in Table 2.4. The invalid CI regions (CI < 0 or CI > 1) have shrunk as the sample sizes n increased, but even taking the very large sample size of one million, the invalid CI regions of proportions

Table 2.4: Invalid CIs of proportions using the normal distribution

α = 0.05					
Sample Size = 10000		*Sample Size = 100000*		*Sample Size = 1000000*	
Proportions	*Upper CIs*	*Proportions*	*Upper CIs*	*Proportions*	*Upper CIs*
0.99997	1.000004	0.99997	1.000004	0.999997	1.0000004
0.99998	1.0000077	0.99998	1.0000077	0.999998	1.0000008
0.99999	1.0000096	0.99999	1.0000096	0.999999	1.000001

α = 0.01					
Sample Size = 10000		*Sample Size = 100000*		*Sample Size = 1000000*	
Proportions	*Upper CIs*	*Proportions*	*Upper CIs*	*Proportions*	*Upper CIs*
0.9994	1.0000309	0.99994	1.0000031	0.999994	1.0000003
0.9995	1.0000759	0.99995	1.0000076	0.999995	1.0000008
0.9996	1.0001151	0.99996	1.0000115	0.999996	1.0000012
0.9997	1.0001461	0.99997	1.0000146	0.999997	1.0000015
0.9998	1.0001643	0.99998	1.0000164	0.999998	1.0000016
0.9999	1.0001576	0.99999	1.0000158	0.999999	1.0000016

α = 0.001					
Sample Size = 10000		*Sample Size = 100000*		*Sample Size = 1000000*	
Proportions	*Upper CIs*	*Proportions*	*Upper CIs*	*Proportions*	*Upper CIs*
0.999	1.0000401	0.9999	1.0000041	0.99999	1.0000004
0.9991	1.0000868	0.99991	1.0000087	0.999991	1.0000009
0.9992	1.0001304	0.99992	1.0000131	0.999992	1.0000013
0.9993	1.0001703	0.99993	1.0000171	0.999993	1.0000017
0.9994	1.0002058	0.99994	1.0000206	0.999994	1.0000021
0.9995	1.0002357	0.99995	1.0000236	0.999995	1.0000024
0.9996	1.000258	0.99996	1.0000258	0.999996	1.0000026
0.9997	1.0002699	0.99997	1.000027	0.999997	1.0000027
0.9998	1.0002653	0.99998	1.0000265	0.999998	1.0000027
0.9999	1.0002291	0.99999	1.0000229	0.999999	1.0000023

still exist.

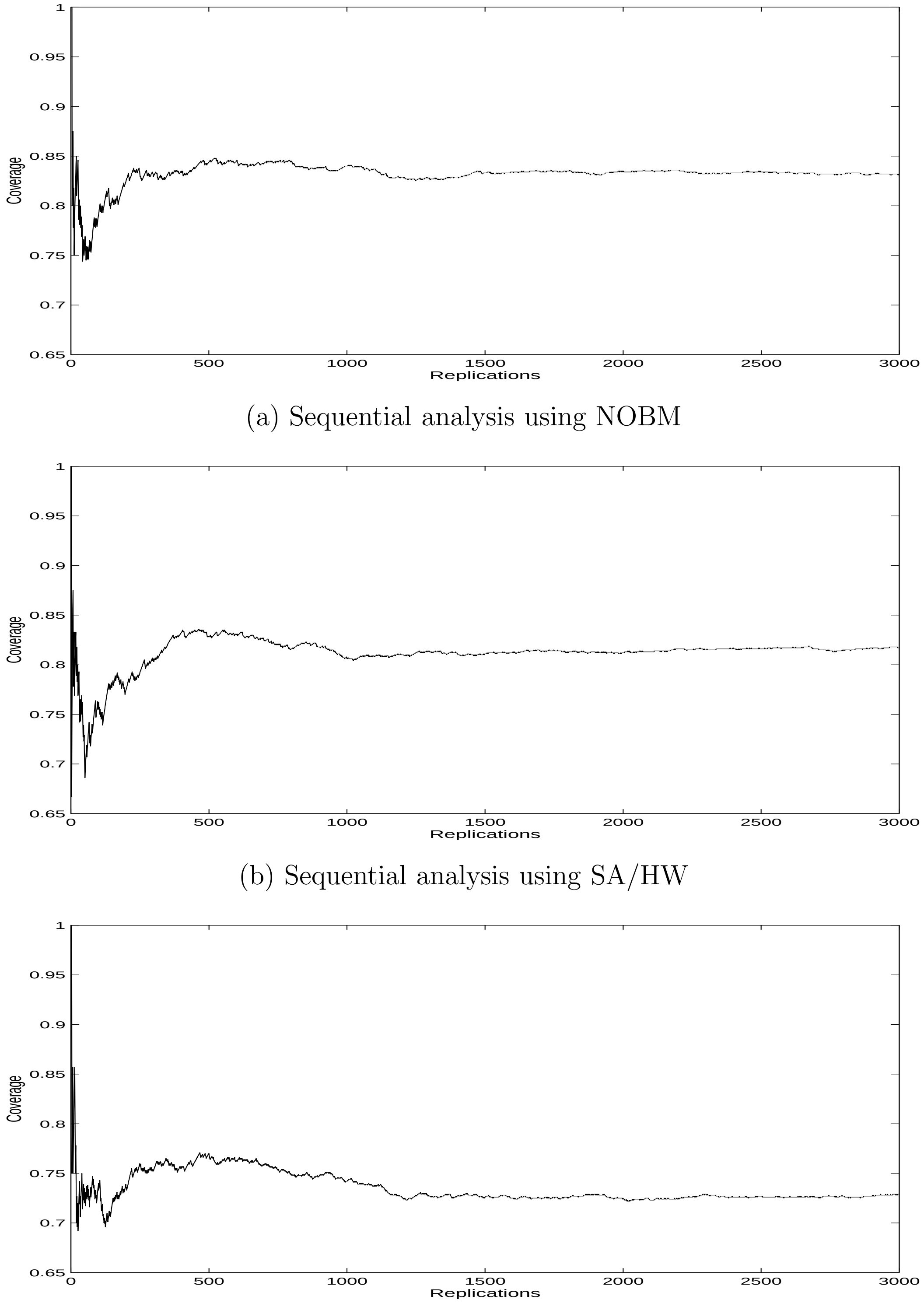
Figure 2.4 and Table 2.4 confirm that interval estimators of proportions based on the *arcsin* transformation and the F distribution never exceed the practical lower and upper limits of the CIs. However, it can be seen that the lower and upper limits of the interval estimator of proportions based on the normal distribution can exceed the lower limit of 0.0 and the upper limit of 1.0, making it inappropriate for simulation coverage analysis.

2.5 Rules for Experimental Coverage Analysis of Sequential Simulation

As recently argued in [134], only sequential coverage analysis can lead to credible final conclusions regarding the quality of any method of simulation output analysis. In the past, as discussed in Section 2.1, coverage analysis has been performed with a fixed number of replications, for example, between 10 - 200. Experimental results, such as those in Figure 2.5, clearly reveal the high initial instability of coverage for the three different methods of mean value analysis: NOBM, SA/HW, and RCs, respectively. To avoid taking the final result from this region, coverage analysis has to be conducted over a sufficiently large sample of data (in this case, after sequential simulation is repeated sufficiently many times).

The final results in Figure 2.5 are far from the assumed confidence level of 0.95 since they include very short simulation runs, which produce heavily biased results. To improve the final coverage to the assumed level of confidence, some rules for experimental coverage analysis including the sequential approach have been proposed in [134]. In this section, we improve those stopping rules by adding one more rule. This is an enhanced version of sequential coverage analysis, based on the F distribution, which leads to more accurate interval estimators of proportions as shown in Section 2.4; see also [102].

Any sequential simulation experiment may stop after too few simulation observations have been collected, if, by chance, the stopping criteria has been



(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 2.5: Convergence of coverage ($M/M/1/\infty$ at $\rho = 0.9$, confidence level of 0.95)

temporarily satisfied. This happens in real simulation experiments from time to time and can make estimates of coverage unreliable. Therefore, we definitely have to make sure a simulation runs long enough and is not accidentally stopped to avoid taking results from abnormally short simulation runs. This will be investigated in Chapter 3.

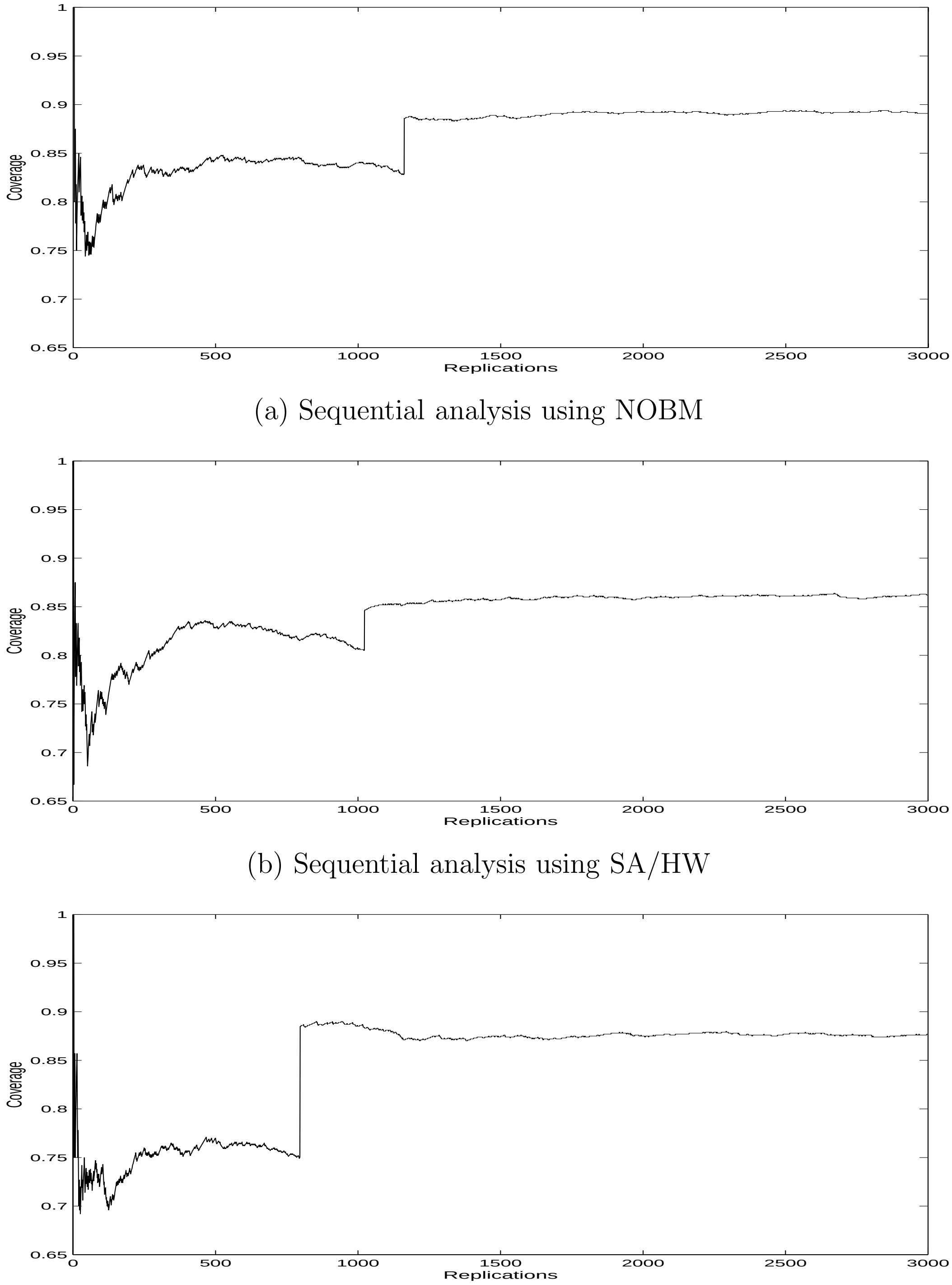
Another practical observation is that when studying coverage of a given method of simulation output data analysis with a range of different conditions (for example, at different traffic levels, in the case of queueing processes) special effort has to be made to ensure that the resulting absolute widths of CIs of coverage are comparable. With high traffic intensities it is necessary to decrease the maximum permitted relative statistical error in sequential coverage analysis, otherwise the final results of simulation are inconclusive since the widths of CIs of coverage are not the same; see Figure 2.1 and [152].

As reported in [134], significant improvements in the final value of the coverage for the three methods of simulation output data analysis (NOBM, SA/HW, and RCs) in the $M/M/1/\infty$ queueing system have been clearly observed after discarding all unreliable simulation results coming from the 'too short' simulation runs; see Figure 2.6. (Simulation runs shorter than a threshold of mean run-length minus one standard deviation of run-lengths were classified as 'too short'.) Comparing Figures 2.5 and 2.6, it is clear that we can draw better conclusions regarding the quality of a given method of simulation output analysis by discarding non-representative simulation runs. In all these cases, however, the final coverage is still far away from the required confidence level of 0.95.

2.5.1 Rules for Experimental Coverage Analysis

On the basis of exhaustive experimental analysis, the rules for the proper experimental analysis of the coverage of sequential steady-state interval estimators, originally formulated in [134], are improved with the addition of an interval estimator based on the F distribution [106], [108]. These are as follows:

- **Rule 1:** Coverage should be analysed sequentially, i.e. analysis of coverage should be stopped when the *absolute precision* of the estimated



(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 2.6: Coverage convergence after discarding 'too short' simulation runs. Filtering started when at least 200 runs having invalid CIs were collected ($M/M/1/\infty$, load = 0.9, confidence level of 0.95)

coverage satisfies a specified level which is sufficiently small.

- **Rule 2:** An estimate of coverage has to be calculated from a representative sample of data, so the coverage analysis can start only after a minimum number of 'bad' CIs[7] have been recorded.

- **Rule 3:** Results from simulation runs that are clearly too short should not be taken into account.

- *An interval estimator which is based on the F distribution of coverage should be used to ensure that the sequential analysis of coverage produces realistic estimates.*

Experimental results of these rules applied to the three sequential methods: NOBM, SA/HW, and RCs, for studying the quality of the final steady-state interval estimators of mean values are presented in Section 2.5.2.

2.5.2 Experimental Results

All experimental results of our sequential coverage analysis of the three sequential methods[8]: NOBM, SA/HW, and RCs, applied to estimate the steady-state means, were obtained assuming that the required statistical error of the final result was 1% or better, at a confidence level of 0.95. Each replication was stopped at the required statistical error of 10% or better.

As justified in [134], one can clearly see that the sequential coverage analysis, with filtering of 'too short' simulation runs and with a requirement of a minimum number of bad CIs, produces more reliable results. Therefore, in a practical implementation of *Rules 1 - 3*, we assume that for data to be representative for coverage analysis, a minimum of 200 bad CIs have to be recorded before sequential analysis can commence, and the results from all simulation

[7] A bad CI means a CI that does not cover the theoretical value of the estimated parameter.

[8] The theoretical bases of these three methods of simulation output data analysis, and sequential implementations of the first two methods, follow exactly the procedures specified in [128]. The last method follows the procedures described in Chapter 4.

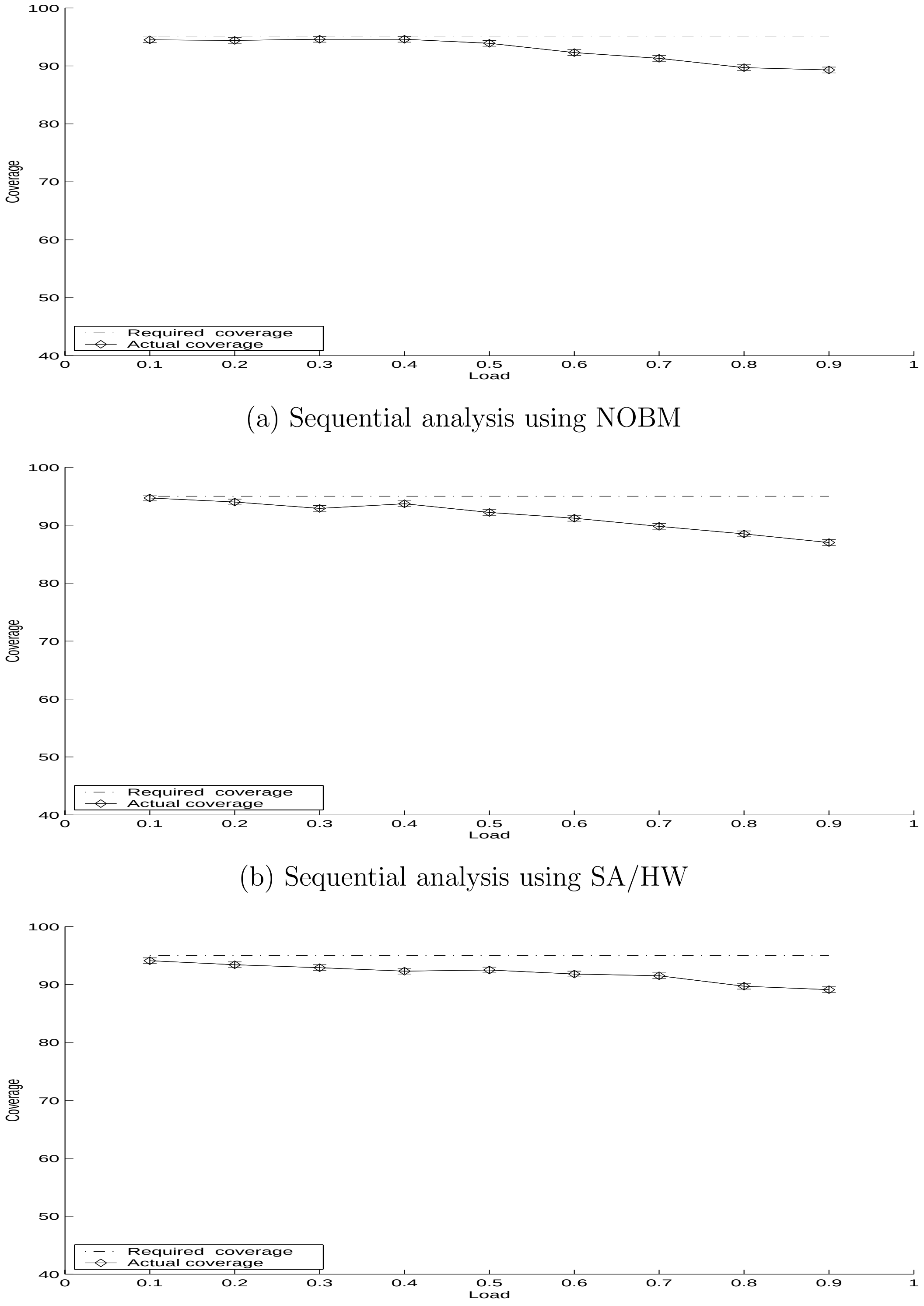
runs shorter than a threshold (one standard deviation below the mean of the simulation run-lengths) should be discarded. Removing the statistical *noise* introduced by 'too short' and unrepresentative simulation runs improves the conclusions we can make about the quality of a given method of simulation output data analysis.

Experimental results for each method when estimating the mean response time, obtained by applying some principles of the sequential coverage analysis discussed in the previous section, are depicted in Figure 2.7 ($M/M/1/\infty$), Figure 2.8 ($M/D/1/\infty$) and Figure 2.9 ($M/H_2/1/\infty$), respectively. The results of the $M/M/1/\infty$ queueing system alone show that all three methods produce a similar (acceptable) coverage, particularly in lightly loaded traffic. However, as the traffic intensities increase, coverage for all methods drops quite far away from the required confidence level of 0.95.

The numerical results of the three methods for the three queueing systems, except the RCs method in the $M/D/1/\infty$ queueing system, show a similar trend to the results reported in [116] and [134], even though the estimated parameters and the assumed statistical errors are different[9]. The difference is that the final half-widths of the CIs at all traffic levels are exactly the same, thus better conclusions can be drawn from these results.

In the case of the RCs method in the $M/D/1/\infty$ queueing system (see Figure 2.8 (c)), the final coverage results in heavily loaded traffic are much better than in lightly loaded traffic, unlike other methods in other queueing systems. In lightly loaded traffic the response times are almost deterministic, since the waiting time in the queue is almost zero and the service time is deterministic. This means that the lengths of collected RCs are often 'too short' (each RC frequently collected only one or two observations). The sequential simulation can stop after only two RCs are observed, since the variance of the response times is very small. In such a case, one can hardly produce valid CIs, since each RC has so few observations. Therefore, the final poor coverage in lightly

[9] The estimated parameters in [116] and [134] are the mean waiting time in the queue. The assumed statistical errors for each replication and coverage analysis are both 5% or better.

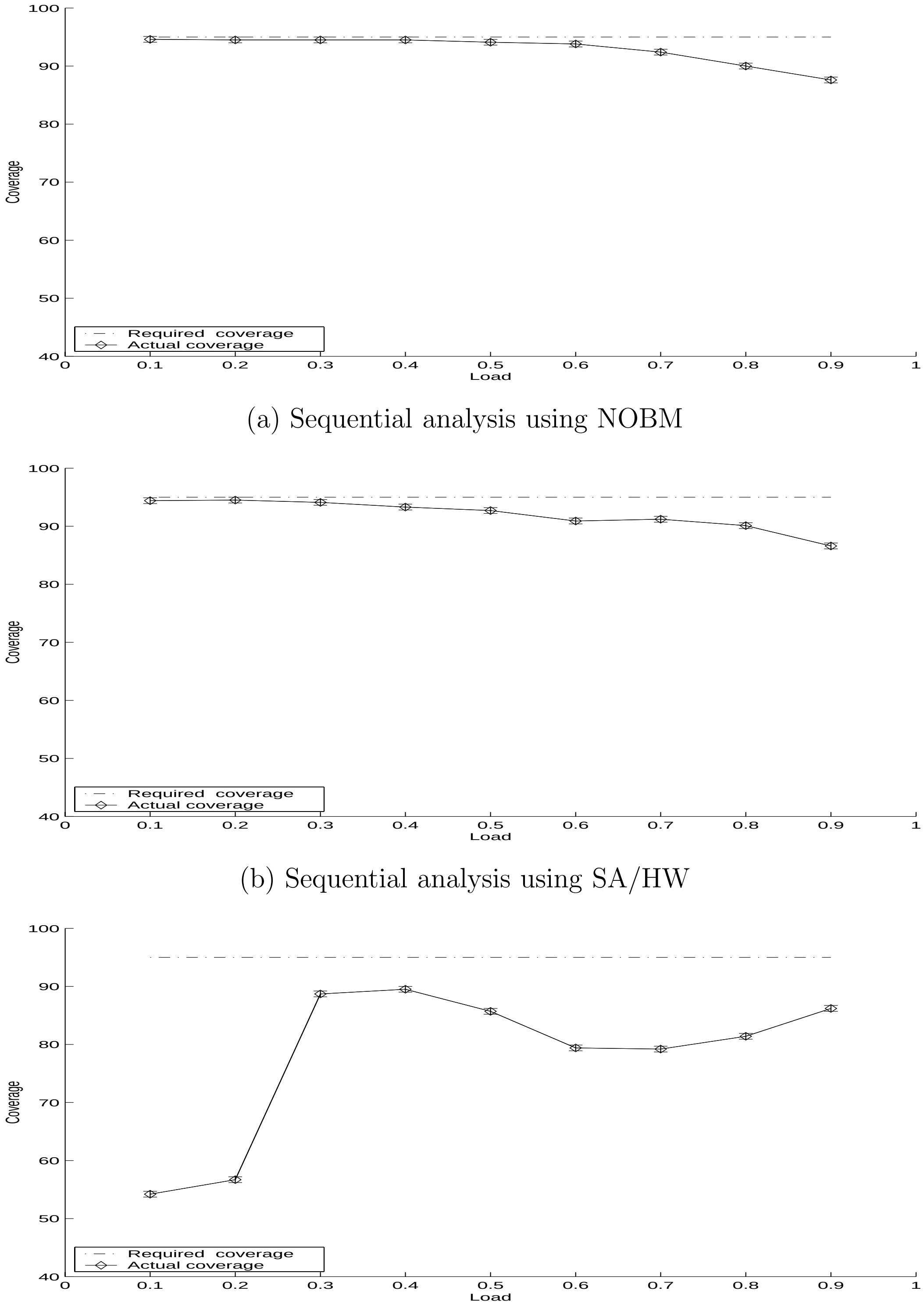


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 2.7: Sequential coverage analysis using F distribution ($M/M/1/\infty$)

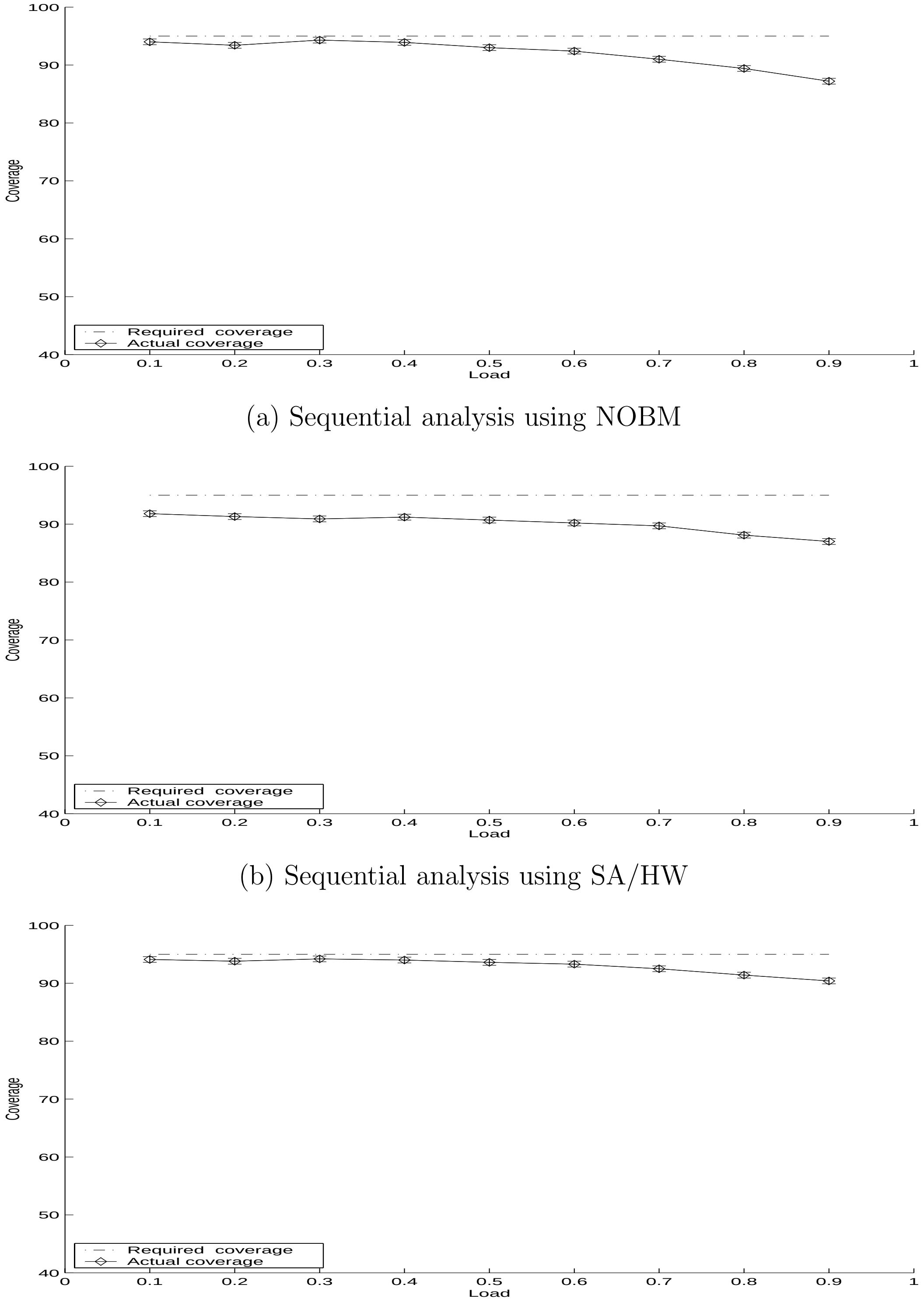


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 2.8: Sequential coverage analysis using F distribution $(M/D/1/\infty)$



(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 2.9: Sequential coverage analysis using F distribution ($M/H_2/1/\infty$)

loaded traffic is inevitable, since it is caused by the theoretical properties of the RCs method and the $M/D/1/\infty$ queueing system.

There is also little sense in discarding 'too short' simulation runs. Not many observations are required even theoretically when estimating the mean response time[10], and simulation runs after filtering 'too short' simulation runs with the threshold (of mean run-length minus one standard deviation of run-lengths) are still not very long. Therefore, the coverage analysed using the RCs method in lightly loaded traffic of the $M/D/1/\infty$ queueing system is much worse than the other methods. A possible solution for this phenomenon will be discussed in Chapter 4.

Ideally, the CIs of coverage should contain the confidence level assumed for the final results [156]. However, the final coverage of each method is still far from the required level, especially in highly correlated systems. The reason for the poor coverage, especially in the sequential RCs method, will be fully investigated in Chapter 4.

2.6 Conclusions

In a simulation, experimental studies of coverage analysis are still required to assess the quality of the practical implementations of the methods of simulation output data analysis used to determine CIs in sequential stochastic simulations. In this chapter, we have studied three interval estimators of proportions, in the context of their applications in sequential coverage analysis. These estimators (based on the normal distribution approximation, the *arcsin* transformation and the F distribution) were applied to the sequential coverage analysis of the SA/HW method of analysis of steady-state mean response times, in simulations of the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems. Although the numerical results of coverage analysis show that they are basically equivalent, there are some concerns about their validity. Estimators based on the F distribution have been found to be more accurate and

[10] See Appendix F for a discussion of the theoretically required run-length for stationary queueing systems.

appropriate for use in coverage studies, especially if a higher value of confidence level is assumed.

CI estimators for proportions using the (symmetric) normal approximation have been commonly used for coverage analysis of simulation output data even though alternative estimators of (asymmetric) CIs for proportions have been proposed in the past. This is probably because the normal approximation is easier to calculate than other interval estimators. However, current computing technology can now deal with alternative estimators. Even CIs for coverage analysis based on the F distribution can be calculated easily by a standard computer.

On the basis of our experimental studies, we enhanced some basic rules for the proper experimental coverage analysis of sequential steady-state simulations. The numerical results of the sequential coverage analysis for the three methods: NOBM, SA/HW, and RCs, in simulations of the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems, by applying these proposed rules were also presented. In general, the final coverage of each method is still far from the required level, especially in highly correlated systems.

Chapter 3

A PROBLEM OF TOO SHORT RUNS IN SEQUENTIAL STEADY-STATE SIMULATION

3.1 Introduction

Sequential simulation is recognised as the only practical approach allowing control of the statistical error of the final results of a stochastic simulation. The accuracy of estimates is assessed along a sequence of consecutive checkpoints. Among the possible stopping criteria, probably the most commonly used is the *relative statistical error*, defined as the ratio of the half-width of the CIs and the point estimate of an analysed performance measure (see Equation (1.2) in Section 1.2). The advantage of using a relative measure of statistical error is that the simulator does not need to know the magnitude of the point estimates of the performance measures. Without any prior knowledge of the run-length of the simulation, the sequential approach is able to guarantee that the final results of the simulation always have the desired level of confidence.

In any correctly implemented simulation, the width of a CI of the simulation result will tend to be reduced as the number of observations increases, i.e. with the duration of a simulation. For example, to obtain the estimate of the mean, with a relative statistical error of 5% or better, at 0.95 confidence level and assuming the central limit theorem, the stopping rule, with the relative statistical error of the CI shown in Equation (1.2), halts the simulation after n observations are collected, i.e.

$$\frac{1.96}{0.05} \leq \frac{\bar{X}(n)}{\hat{\sigma}[\bar{X}(n)]}, \tag{3.1}$$

where $\hat{\sigma}^2[\bar{X}(n)]$ is the unbiased estimator of the variance of $\bar{X}(n)$. Finding this unbiased estimator is a major analytical problem in stochastic simulation.

Typically, in long simulation runs, the convergence of the relative statistical error to its threshold value is very slow, but persistent, as shown in Figure 3.1. However, we can also see the sudden increase or decrease of the relative statistical error in Figure 3.1. This is caused by the fact that the variance estimated from observations collected during the last two checkpoints sometimes unexpectedly increases or decreases. Consequently, a problem of sequential simulation with such a stopping rule is that the inherently random nature of simulation output data generated during any stochastic simulation can cause an accidental, temporary satisfaction of the stopping rule because of a very

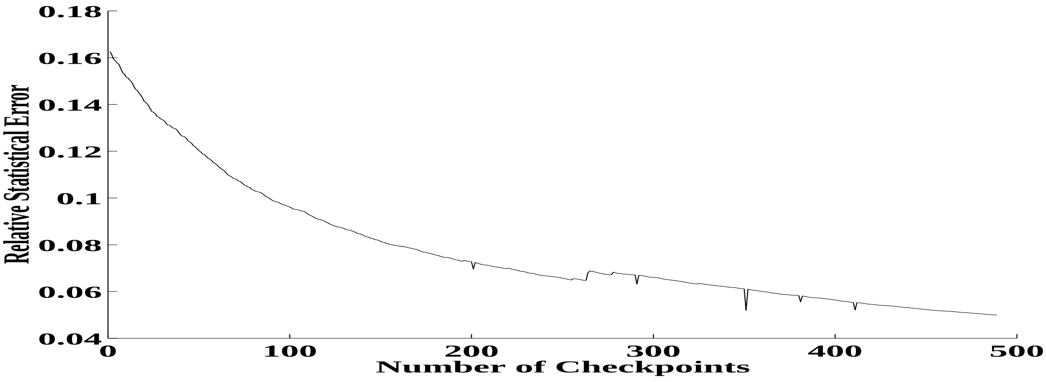


Figure 3.1: Convergence on a relative statistical error of 5% for the sequential method of NOBM (when estimating the mean response time in the $M/M/1/\infty$ queueing system at load = 0.9, checkpoints spaced linearly: 1,250 observations between two checkpoints)

small variance estimate [105], [107]. Such prematurely finished simulations can produce very inaccurate estimates. Experimental evidence of this phenomenon, and the resulting significant degradation of the coverage of the final results of the simulation, are documented in Section 3.2.

We propose and compare some simple heuristic rules that can offer a possible solution to this problem. Their effectiveness is quantitatively assessed on the basis of the final results of coverage analysis of three sequential estimators of mean values in the context of a steady-state simulation. A few 'rules of thumb' to improve the coverage of the final CIs in practical applications of fully automated sequential steady-state simulations are discussed in Section 3.3. The performance evaluations of the rules, in terms of coverage, are presented in Section 3.4. The theoretical and empirical run-lengths required for some queueing models are compared in Section 3.5. In Section 3.6, the relationship between the coverage and the run-length is discussed. Finally, conclusions are presented in Section 3.7.

3.2 A Problem of Early Stopping: Experimental Evidence

A problem faced in practical applications of sequential steady-state simulation is that an assumed stopping criterion, for example, one based on the relative statistical error, can be accidently satisfied too early, giving very inaccurate estimates of the analysed parameters. This happens due to the random nature of the fluctuations in the estimated relative statistical error during the stochastic simulation; also see, for example [130]. Therefore, whatever relative statistical errors and simulation output data analysis methods are applied, abnormally short simulation runs can always occur in sequential simulation practice.

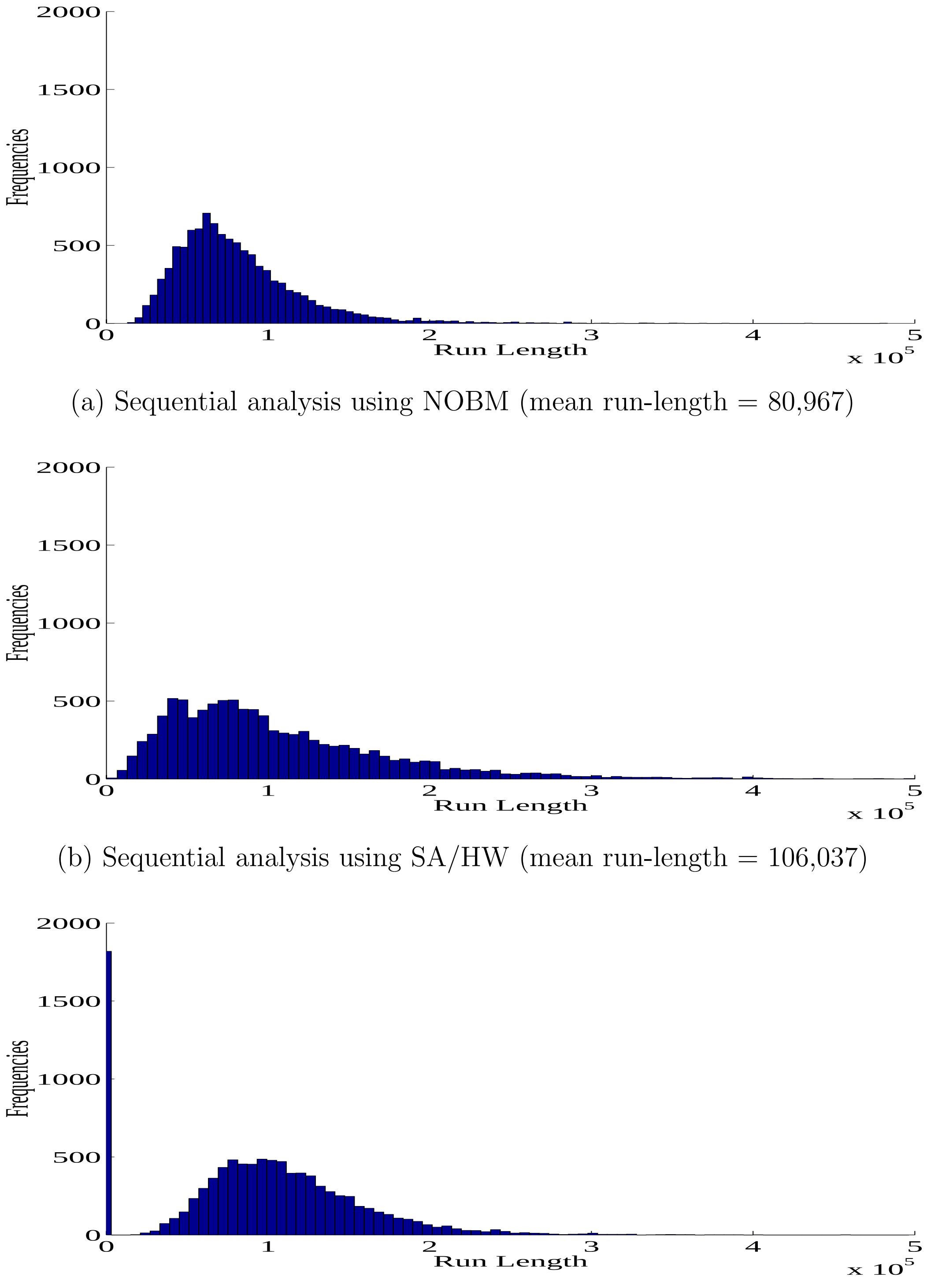
At least a dozen methods have been proposed for estimating the CIs of the autocorrelated time-series of observations collected to study the steady-state means. A survey of such methods used until 1990 can be found in [128]. Newer methods have appeared in [52] and [68]. In Chapter 3, we restrict our discussion to three methods of sequential mean value analysis: NOBM, SA/HW, and RCs. A detailed discussion of these three methods of simulation output data analysis is given in Appendix B.

Since experimental investigation of the consequences of prematurely finished simulation runs requires that the exact values of the analysed parameters are known, we use the results obtained from the sequential steady-state mean value simulation of three analytically tractable queueing systems: $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$. These queueing systems are widely used as reference models in research on methods of simulation output data analysis, since they have different degrees of autocorrelation of data in output sequences and require relatively long simulation runs to achieve a satisfactorily low level of error when estimating the mean response time or mean waiting time in the queue at high traffic level [158].

Figure 3.2 ($M/M/1/\infty$), Figure 3.3 ($M/D/1/\infty$), and Figure 3.4 ($M/H_2/1/\infty$) give histograms of the run-lengths of 10,000 independent simulation replications, when estimating the mean response time in the corresponding queueing system at load of 0.9 with a relative statistical error of 10% at the confidence level of 0.95. Note that the simulation run-lengths for the three methods: NOBM, SA/HW, and RCs, were measured by the number of collected observations to facilitate comparisons. The empirical mean run-lengths of 10,000 sequential steady-state simulations obtained using each method for the three queueing systems are presented in Table 3.1.

Table 3.1: Mean run-lengths of 10,000 sequential steady-state simulations (when estimating the mean response time at load = 0.9 with a relative statistical error of 10% at a confidence level = 0.95)

	$M/M/1/\infty$	$M/D/1/\infty$	$M/H_2/1/\infty$
Using *NOBM*	80,967	39,129	281,427
Using *SA/HW*	106,037	44,845	403,492
Using *RCs*	92,959	26,105	373,401

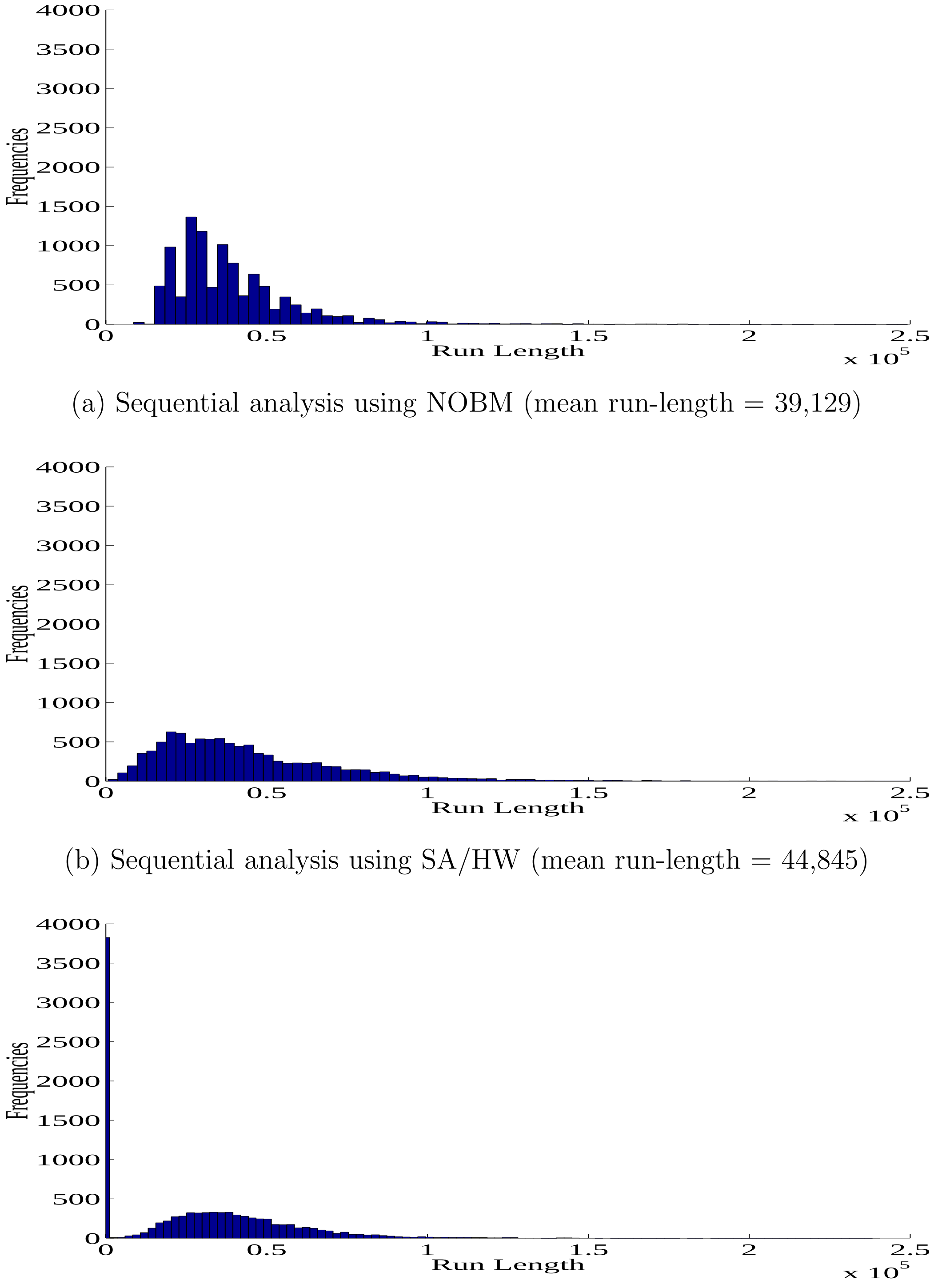


(a) Sequential analysis using NOBM (mean run-length = 80,967)

(b) Sequential analysis using SA/HW (mean run-length = 106,037)

(c) Sequential analysis using RCs (mean run-length = 92,959)

Figure 3.2: Histogram of simulation run-lengths ($M/M/1/\infty$, load = 0.9, 10,000 replications)

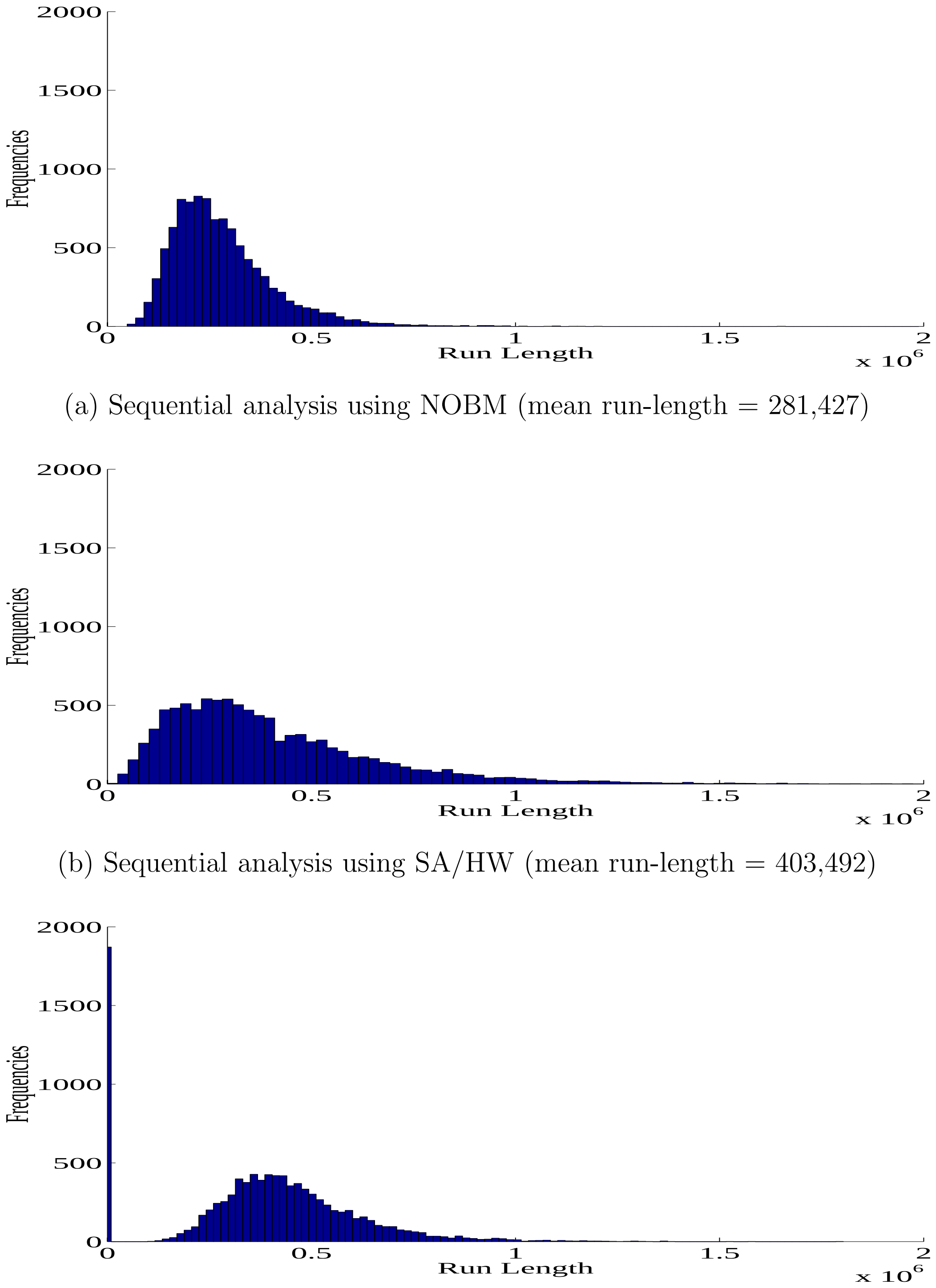


(a) Sequential analysis using NOBM (mean run-length = 39,129)

(b) Sequential analysis using SA/HW (mean run-length = 44,845)

(c) Sequential analysis using RCs (mean run-length = 26,105)

Figure 3.3: Histogram of simulation run-lengths ($M/D/1/\infty$, load = 0.9, 10,000 replications)



(a) Sequential analysis using NOBM (mean run-length = 281,427)

(b) Sequential analysis using SA/HW (mean run-length = 403,492)

(c) Sequential analysis using RCs (mean run-length = 373,401)

Figure 3.4: Histogram of simulation run-lengths ($M/H_2/1/\infty$, load = 0.9, 10,000 replications)

The theoretically required simulation run-lengths when estimating the mean response time, at a load of 0.9 with $\epsilon_{max} \cdot 100\% = 10\%$ as the upper level of the acceptable relative statistical error of the final results, at a confidence level of 0.95, are 145,596 observations ($M/M/1/\infty$), 60,557 observations ($M/D/1/\infty$), and 546,971 observations ($M/H_2/1/\infty$); see Appendix F for a detailed discussion of how to calculate the theoretical run-lengths of sequential steady-state simulations. Comparing recorded run-lengths of the simulation in Figure 3.2 ($M/M/$ $1/\infty$), Figure 3.3 ($M/D/1/\infty$), and Figure 3.4 ($M/H_2/1/\infty$) with the theoretical simulation run-lengths, we can see that many runs do not collect enough observations.

We can also see the spikes only in the method of sequential RCs; see Figures 3.2 (c), 3.3 (c), and 3.4 (c). The ranges of those spikes are 2 - 336 observations (for $M/M/1/\infty$), 2 - 160 observations (for $M/D/1/\infty$), and 2 - 4,037 observations (for $M/H_2/1/\infty$). Many runs are much shorter or even collecting as few as two observations. This phenomenon will be fully explored in Chapter 4.

Analyses of the random run-lengths of the sequential steady-state simulation for NOBM, SA/HW, and RCs are presented in Tables 3.2 - 3.4 ($M/M/1/\infty$), Tables 3.5 - 3.7 ($M/D/1/\infty$), and Tables 3.8 - 3.10 ($M/H_2/1/\infty$). Each of the results was obtained from 10,000 independent replications of the sequential steady-state simulation. Following the proposal in [134], we have classified a simulation as 'too short' if its run-length was shorter than a threshold, which is the mean simulation run-length minus one standard deviation of run-lengths. The threshold values of the minimum acceptable run-lengths of simulations and the overall experimental mean simulation run-lengths are given in the last two columns. The second and fourth columns also give, respectively, the absolute and the relative number of 'too short' simulation runs over the total number (10,000 replications) of simulations executed at each load level of each queueing system.

It can be seen that the NOBM method (Tables 3.2, 3.5, and 3.8) produces mean run-lengths and threshold values much higher than the SA/HW and RC methods, especially when the queueing systems are lightly loaded, since the final acceptable batch size for NOBM was determined after 10,000 observations

Table 3.2: Sequential method of NOBM from 10,000 replications (when estimating the mean response time from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error ≤ 10%)

ρ	*Number of too short runs*	*Coverage of too short runs*	*Probability of being too short*	*Threshold for filtering*	*Mean of run-lengths*
0.1	0	N/A	0.0%	8471	11823
0.2	0	N/A	0.0%	8567	11888
0.3	0	N/A	0.0%	8424	11967
0.4	0	N/A	0.0%	8451	12221
0.5	0	N/A	0.0%	8356	12538
0.6	0	N/A	0.0%	8175	13242
0.7	0	N/A	0.0%	8893	15586
0.8	593	50.6%	5.9%	13318	24826
0.9	1017	35.1%	10.2%	41596	80967

Table 3.3: Sequential method of SA/HW from 10,000 replications (when estimating the mean response time from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error ≤ 10%)

ρ	*Number of too short runs*	*Coverage of too short runs*	*Probability of being too short*	*Threshold for filtering*	*Mean of run-lengths*
0.1	0	N/A	0.0%	1345	1725
0.2	1	100.0%	0.01%	1392	2006
0.3	571	86.0%	5.7%	1549	2493
0.4	1749	77.9%	17.5%	1839	3302
0.5	1069	69.5%	10.7%	2374	4665
0.6	1138	64.1%	11.4%	3356	7277
0.7	1101	53.8%	11.0%	5383	12701
0.8	1000	47.9%	10.0%	10461	27809
0.9	928	38.8%	9.3%	34933	106037

Table 3.4: Sequential method of RCs from 10,000 replications (when estimating the mean response time from the $M/M/1/\infty$ queueing system at a confidence level $= 0.95$ with a statistical error $\leq 10\%$)

ρ	*Number of too short runs*	*Coverage of too short runs*	*Probability of being too short*	*Threshold for filtering*	*Mean of run-lengths*
0.1	648	14.7%	6.5%	343	511
0.2	776	17.3%	7.8%	456	738
0.3	893	18.9%	8.9%	641	1101
0.4	941	19.3%	9.4%	927	1685
0.5	1022	17.5%	10.2%	1448	2743
0.6	1064	11.8%	10.6%	2388	4738
0.7	1151	10.2%	11.5%	4482	9378
0.8	1302	6.1%	13.0%	9804	22552
0.9	1850	5.1%	18.5%	31333	92959

Table 3.5: Sequential method of NOBM from 10,000 replications (when estimating the mean response time from the $M/D/1/\infty$ queueing system at a confidence level $= 0.95$ with a statistical error $\leq 10\%$)

ρ	*Number of too short runs*	*Coverage of too short runs*	*Probability of being too short*	*Threshold for filtering*	*Mean of run-lengths*
0.1	0	N/A	0.0%	8526	12043
0.2	0	N/A	0.0%	8520	11967
0.3	0	N/A	0.0%	8473	11986
0.4	0	N/A	0.0%	8438	12099
0.5	0	N/A	0.0%	8389	12313
0.6	0	N/A	0.0%	8307	12685
0.7	0	N/A	0.0%	8466	13517
0.8	0	N/A	0.0%	9470	17024
0.9	532	42.1%	5.3%	19070	39129

Table 3.6: Sequential method of SA/HW from 10,000 replications (when estimating the mean response time from the $M/D/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%)

ρ	*Number of too short runs*	*Coverage of too short runs*	*Probability of being too short*	*Threshold for filtering*	*Mean of run-lengths*
0.1	1300	94.6%	13.0%	1923	2199
0.2	1110	93.9%	11.1%	1653	1811
0.3	888	92.8%	8.9%	1575	1708
0.4	222	91.4%	2.2%	1512	1701
0.5	0	N/A	0.0%	1423	1851
0.6	39	89.7%	0.4%	1483	2458
0.7	1076	63.3%	10.8%	1975	4218
0.8	928	46.1%	9.3%	3920	10225
0.9	873	34.2%	8.7%	14557	44845

Table 3.7: Sequential method of RCs from 10,000 replications (when estimating the mean response time from the $M/D/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%)

ρ	*Number of too short runs*	*Coverage of too short runs*	*Probability of being too short*	*Threshold for filtering*	*Mean of run-lengths*
0.1	4323	0.0%	43.2%	2	7
0.2	7322	13.2%	73.2%	9	11
0.3	7323	11.5%	73.2%	20	23
0.4	6861	9.8%	68.6%	43	55
0.5	6299	6.2%	63.0%	87	138
0.6	5323	2.6%	53.2%	163	381
0.7	4325	1.6%	43.3%	263	1187
0.8	3635	2.1%	36.4%	215	4412
0.9	3830	2.9%	38.3%	198	26105

Table 3.8: Sequential method of NOBM from 10,000 replications (when estimating the mean response time from the $M/H_2/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%)

ρ	*Number of too short runs*	*Coverage of too short runs*	*Probability of being too short*	*Threshold for filtering*	*Mean of run-lengths*
0.1	0	N/A	0.0%	8371	12308
0.2	0	N/A	0.0%	8703	13007
0.3	1240	79.6%	12.4%	10607	15131
0.4	1171	73.3%	11.7%	13595	19200
0.5	1084	67.3%	10.8%	17943	25334
0.6	1346	64.2%	13.5%	24530	34924
0.7	1212	59.8%	12.1%	35658	52293
0.8	1207	52.8%	12.1%	59399	93866
0.9	1084	38.6%	10.8%	153571	281427

Table 3.9: Sequential method of SA/HW from 10,000 replications (when estimating the mean response time from the $M/H_2/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%)

ρ	*Number of too short runs*	*Coverage of too short runs*	*Probability of being too short*	*Threshold for filtering*	*Mean of run-lengths*
0.1	1306	69.4%	13.1%	3546	6863
0.2	1228	67.7%	12.3%	5432	10955
0.3	1160	66.2%	11.6%	7586	15768
0.4	1201	64.3%	12.0%	10180	21750
0.5	1151	61.4%	11.5%	13785	30438
0.6	1176	61.3%	11.8%	18865	42893
0.7	1109	56.3%	11.1%	27939	67654
0.8	1141	52.9%	11.4%	49613	126815
0.9	972	43.7%	9.7%	136367	403492

Table 3.10: Sequential method of RCs from 10,000 replications (when estimating the mean response time from the $M/H_2/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%)

ρ	*Number of too short runs*	*Coverage of too short runs*	*Probability of being too short*	*Threshold for filtering*	*Mean of run-lengths*
0.1	760	20.7%	7.6%	3341	4919
0.2	744	14.8%	7.4%	5766	8678
0.3	720	17.9%	7.2%	8858	13183
0.4	719	15.6%	7.2%	12423	18865
0.5	792	19.3%	7.9%	17464	26918
0.6	843	10.7%	8.4%	24088	39184
0.7	946	8.1%	9.5%	35642	62530
0.8	1201	5.5%	12.0%	59181	119924
0.9	1883	4.6%	18.8%	136592	373401

collected. Because the threshold value is high for NOBM, there are often no 'too short' simulation runs; see Tables 3.2, 3.5, and 3.8.

As discussed in Chapter 2, the quality of the final results produced by the 'too short' simulation runs can be assessed by their coverage, i.e. by the experimental frequency with which the final CIs of the results contain the theoretical value of the estimated parameter. In an ideal situation, the coverage should be close to the assumed confidence level. However, a closer look at the statistical analysis of the 'too short' simulation runs reveals that the coverage of the CIs of the simulation results obtained during such a run can be very poor indeed; see the third column in Tables 3.2 - 3.10.

Additionally, we note that the probability of a simulation run-length being 'too short' cannot be ignored; see the fourth column in Tables 3.2 - 3.10. The probability of a run being too short is quite high and the resulting coverage is not at an acceptable level. While this should be of concern in the case of any method considered, the coverage of the CIs in the method of RCs is particularly very low.

Experimental results show how wrong final simulation results obtained from 'too short' simulation runs can be in practice. Such a problem needs to be recognised in practical applications of fully automated sequential steady-state simulations. Therefore, a rule for preventing those 'too short' runs from determining the final results is needed.

3.3 Heuristic Rules for Preventing the Final Results Coming from 'Too Short' Runs

Most methods' implementations of simulation output data analysis run simulations *only once* until the acceptable statistical error is reached. However, as shown in the experimental results in Section 3.2, a single sequential simulation run can be 'too short', leading to erroneous results whichever output data analysis method (NOBM, SA/HW or RCs) is used [105], [107]. All results presented in Section 3.2 were obtained when estimating the mean response time from the sequential steady-state simulation of three analytically tractable queueing systems: $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$. One can also see that over the set of reference models, the problem becomes more critical with heavily loaded queueing systems, or, equivalently, with processes with stronger autocorrelations.

Our results show that it is important, in practical applications, to eliminate 'too short' simulation runs. Fortunately, significant achievements in computing technologies have made CPU time very much cheaper, which makes it possible to obtain reliable results within a reasonable time for very long sequential steady-state simulations, and should also allow a simulation to be repeated several times, producing more credible final results. This is not a new idea. As D. Knuth wrote in 1969 *"... the most prudent policy for a person to follow is to run each Monte Carlo program at least twice, using quite different sources of pseudo-random numbers, before taking the answers of the program seriously"* [85].

In this section, we propose five simple 'rules of thumb' which could help to eliminate the effect of 'too short' simulation runs in sequential steady-state simulations. Those rules are based on two ideas: (i) using only one run of several executed runs (Rules I to III), or (ii) using all runs without discarding any results (Rules IV and V).

Heuristic Rules: I

A simple rule of thumb, which can help to avoid taking 'too short' runs of sequential steady-state simulation into account, can be formulated as follows.

1. Execute R independent replications of a given simulation and record the run-lengths (measured by the size of the sample of simulation output data).

2. Accept the result produced by the longest simulation run only.

Using the results presented in Tables 3.2 - 3.10, one can assess the probability that, having applied Rule I, one would still deal with the final results from a 'too short' simulation run. That is, if one executes R independent replications, $R \geq 1$, and P_{short} is the probability that a simulation run is 'too short', then $(P_{short})^R$ is the probability of all R independent replications belonging to the class of 'too short' simulation runs.

Using the worst examples from the sequential steady-state simulations of Section 3.2, the probabilities of 'too short' runs can be seen in Table 3.11. The probability quickly becomes negligible with an increased number of runs, except for the case of sequential RCs in the $M/D/1/\infty$ queueing system[1].

Heuristic Rules: II

The relative statistical error randomly changes with the number of collected simulation observations, although it tends to reduce until the minimum level

[1] The reason for the high probability of being a 'too short' simulation run in the case of sequential RCs for the $M/D/1/\infty$ queueing system will be discussed in Chapter 4.

Table 3.11: The probability of R independent replications belonging to the class of 'too short' simulation runs (theoretical confidence level = 0.95)

(a) $M/M/1/\infty$ queueing system

Num. of runs	NOBM ($\rho = 0.9$)	SA/HW ($\rho = 0.4$)	RCs ($\rho = 0.9$)
$R = 1$	$0.102^1 = 0.102$	$0.175^1 = 0.175$	$0.185^1 = 0.185$
$R = 2$	$0.102^2 = 0.0104$	$0.175^2 = 0.0306$	$0.185^2 = 0.0342$
$R = 3$	$0.102^3 = 0.0011$	$0.175^3 = 0.0054$	$0.185^3 = 0.0063$
$R = 5$	$0.102^5 = 0.00001$	$0.175^5 = 0.00016$	$0.185^5 = 0.00022$

(b) $M/D/1/\infty$ queueing system

Num. of runs	NOBM ($\rho = 0.9$)	SA/HW ($\rho = 0.1$)	RCs ($\rho = 0.2/0.3$)
$R = 1$	$0.053^1 = 0.053$	$0.130^1 = 0.130$	$0.732^1 = 0.732$
$R = 2$	$0.053^2 = 0.0028$	$0.130^2 = 0.0169$	$0.732^2 = 0.5358$
$R = 3$	$0.053^3 = 0.0001$	$0.130^3 = 0.0022$	$0.732^3 = 0.3922$
$R = 5$	$0.053^5 = $ 4e-7	$0.130^5 = 0.00004$	$0.732^5 = 0.21016$

(c) $M/H_2/1/\infty$ queueing system

Num. of runs	NOBM ($\rho = 0.6$)	SA/HW ($\rho = 0.1$)	RCs ($\rho = 0.9$)
$R = 1$	$0.135^1 = 0.135$	$0.131^1 = 0.131$	$0.188^1 = 0.188$
$R = 2$	$0.135^2 = 0.0182$	$0.131^2 = 0.0172$	$0.188^2 = 0.0353$
$R = 3$	$0.135^3 = 0.0025$	$0.131^3 = 0.0022$	$0.188^3 = 0.0066$
$R = 5$	$0.135^5 = 0.00004$	$0.131^5 = 0.00004$	$0.188^5 = 0.00023$

(or better) of required statistical error is reached. The smaller the reported relative statistical error, the better the accuracy of the final results. Thus, one way of producing the most accurate final result could be to take results from simulation runs with the smallest relative statistical errors. This gives us the following rule:

1. Execute R independent replications of a given simulation and record the final relative statistical error of the results.

2. Accept the result with the smallest relative statistical error only.

Heuristic Rules: III

Wide CIs can produce good coverage, and, conversely, narrow CIs can produce poor coverage. This means that an easy way to guarantee a satisfactory level of coverage with acceptable statistical errors of the final results in the sequential steady-state simulation is to take the results from simulation runs with the widest CIs. Thus, let us consider the following rule:

1. Execute R independent replications of a given simulation and record the final CIs of the results.
2. Accept the result with the widest CI only.

Heuristic Rules: IV

To ensure that the run-length of a sequential steady-state simulation is acceptably close to the required theoretical run-length, one can easily combine a number of results obtained from independent sequential steady-state simulations. This can prevent to take the final results coming from a 'too short' run. We propose the following rule:

1. Execute R independent replications of a given simulation, and record the run-lengths (measured by the size of the sample of simulation output data) and the estimated values.
2. Accept the result produced by combining R results obtained from R independent replications.

Rule IV needs a mean μ and a variance σ^2 of a combined simulation run to construct the combined CI. The mean value of a combined simulation should be calculated by weighting the R simulation runs, which have different mean values calculated from different sample sizes. The variance of the combined mean can be calculated by using an unbiased estimator of σ^2 for pooled samples. The best way of combining several variance estimates calculated from different sample sizes is to average them with their weightings, which equal to their degrees of freedom $(n_i - 1)$, where n_i is the sample size of the i-th independent replication [120].

Suppose one has variance estimates $s_1^2, s_2^2, \cdots, s_I^2$, from I independent samples of size $n_1, n_2, \cdots, n_I$, from populations with a common variance σ^2. The pooled sample variance is calculated by

$$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2 + \cdots + (n_I - 1)s_I^2}{(n_1 - 1) + (n_2 - 1) + \cdots + (n_I - 1)}, \tag{3.2}$$

which is an unbiased estimator of the variance σ^2. This is called the *pooled estimator* of σ^2 because it combines the information from all samples [120]. This formula gives more weight to groups with larger sample sizes.

Heuristic Rules: V

To guarantee a satisfactory level of coverage with acceptable statistical errors of the final results from the sufficiently long sequential simulation, one can simply combine Rules III and IV to obtain a half-width of a CI and mean, respectively. Thus, we propose the following rule:

1. Execute R independent replications of a given simulation.

2. Record the run-lengths (measured by the size of the sample of simulation output data), the estimated values, and the final CIs of the results.

3. Accept the mean value produced by combining R estimated values obtained from R independent replications (Rule IV).

4. Accept a half-width of a CI from a simulation run with the widest CI among R independent replications (Rule III).

5. Construct a CI with the results obtained in 3 and 4.

Comparisons of Heuristic Rules

Our study shows that all final results of coverage obtained by using the three methods: NOBM, SA/HW, and RCs, are far from the required level of confidence, especially for heavily loaded queueing systems; see Figures 2.7 - 2.9 in Chapter 2. This problem has been identified in various methods of simulation output data analysis whose coverage has been so far analysed sequentially, i.e. it characterises various versions of the method of batch means, the method of SA/HW, the method of RCs, and the method based on the standardised time series; see, for example, [102], [121], [122], and [134]. Therefore, one of the ongoing research problems in the area of sequential steady-state simulation is to find a valid method of simulation output data analysis for highly dynamic stochastic processes, for example, heavily loaded queueing systems and telecommunication networks.

The proposed rules are a significant diversion from running an automated sequential simulation only once, even without a pilot run [65]. Note that Rules I to III discard (R - 1) replications and use only one replication to calculate the final results, while Rules IV and V suggest using all R independent replications. Of course, no heuristic rule of thumb can ensure that the final CIs from a stochastic simulation will contain the theoretical value, with a probability equal to the assumed confidence level. However, these heuristic rules may help preventing 'too short' simulations, which are not representative, from being included in the final results.

3.4 Performance Evaluation of the Proposed Heuristic Rules

In this section, we study the effect of Rules I to V on the quality of the final results, in terms of the accuracy and coverage of CIs from the experimental results of sequential steady-state simulations produced by the three methods: NOBM, SA/HW, and RCs, using the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems as the reference simulation models. The mean response time was estimated with $\epsilon_{max} \cdot 100\% = 10\%$ as the upper level of the acceptable relative statistical error of the final results, at a confidence level of 0.95. In each case the final results are averaged 2,000 independent replications. For example, in the case of $R = 5$ replications, we have used a total of 10,000 replications.

Performance Evaluation: Heuristic Rule I

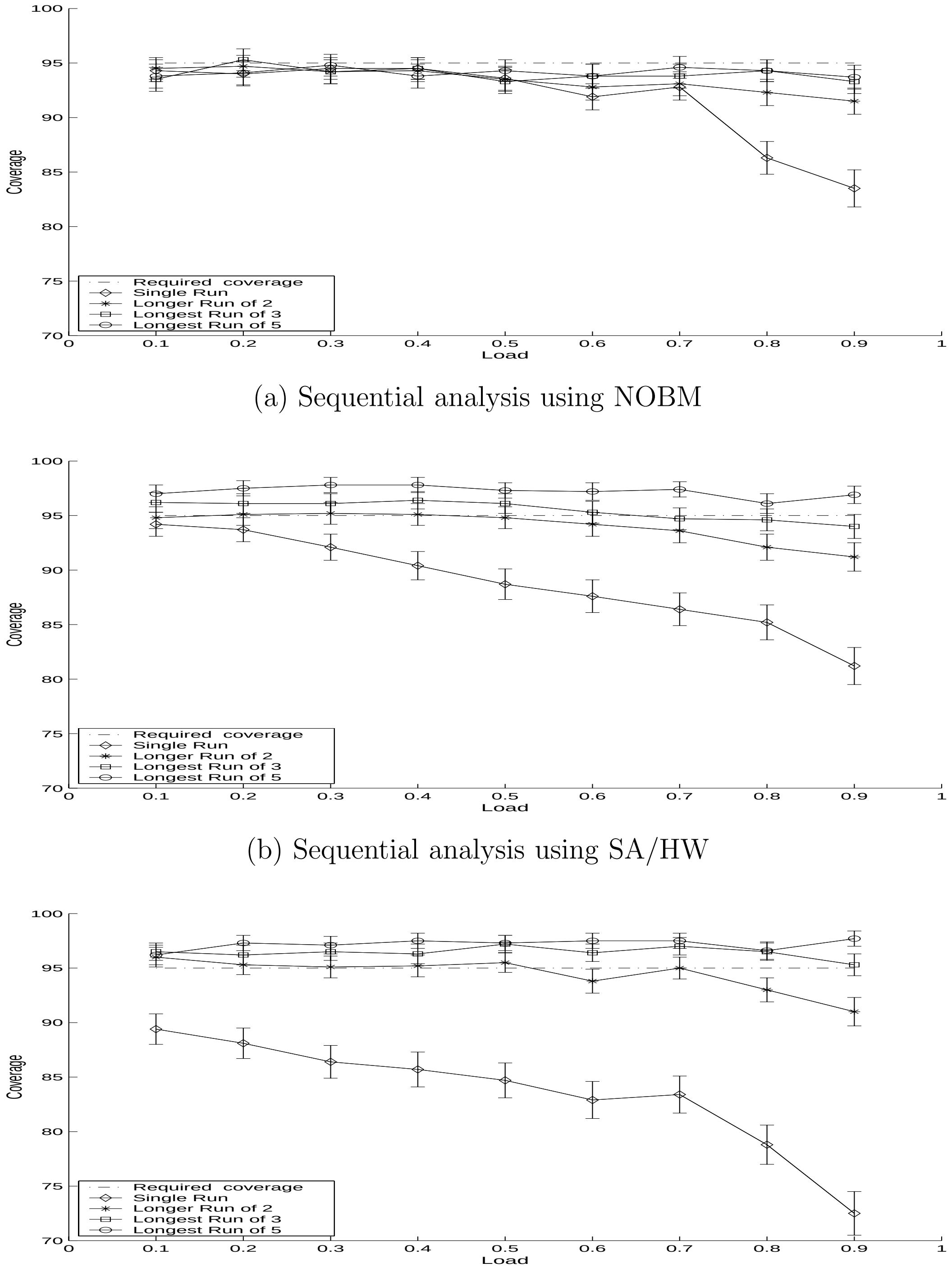
Figure 3.5 ($M/M/1/\infty$), Figure 3.6 ($M/D/1/\infty$), and Figure 3.7 ($M/H_2/1/\infty$) show the application of Rule I, which uses the longest run of the executed R replications; $R = 1$, 2, 3 and 5, for each analysis method. The coverage of the final results clearly shows that Rule I is viable, and the larger R is, the better the quality of the final results.

In fact, in the cases considered, if one always wants to have the final results within a required level of confidence, there is no need to assume that R is larger than 3, since the resulting coverage reaches a satisfactory level at this point (except the sequential analysis using RCs in the $M/D/1/\infty$ queueing system[2]: see Figure 3.6 (c)). This is because as the statistical data of Table 3.11 (a) and (c) show, the probability that the remaining replication is still 'too short', after discarding two shorter replications out of three, drops to 0.007 or less for the RCs method, which is the worst case, in both the $M/M/1/\infty$ and $M/H_2/1/\infty$ queueing systems.

Performance Evaluation: Heuristic Rule II

The results of the coverage obtained by applying Rule II, which takes the most 'accurate' result, i.e., taking the result from the simulation run with the (relatively) smallest relative statistical errors, out of R executed replications; $R =$ 1, 2, 3 and 5, are depicted in Figure 3.8 ($M/M/1/\infty$), Figure 3.9 ($M/D/1/\infty$),

[2] The unacceptable coverage obtained in the sequential analysis using RCs in the $M/D/1/\infty$ queueing system, even assuming that R is larger than 3, is caused by the fact that the probability that the remaining replication, after discarding two shorter replications out of three, is 'too short' is still very high (0.4 or less): see Table 3.11 (b).

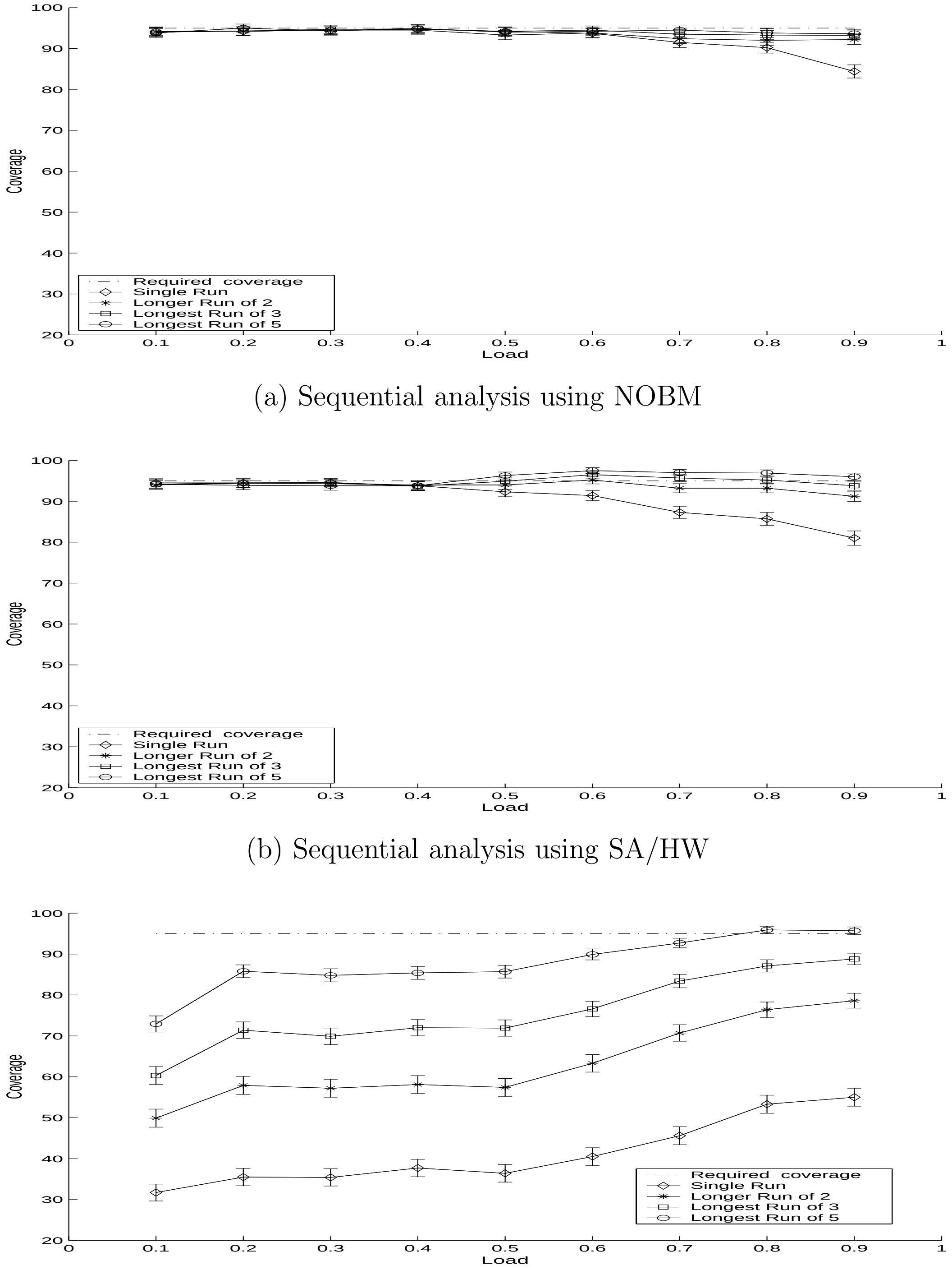


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.5: Coverage of the CIs with Rule I (take the longest of R replications; R = 1, 2, 3 and 5). Estimation of the mean response time in the $M/M/1/\infty$ queueing system

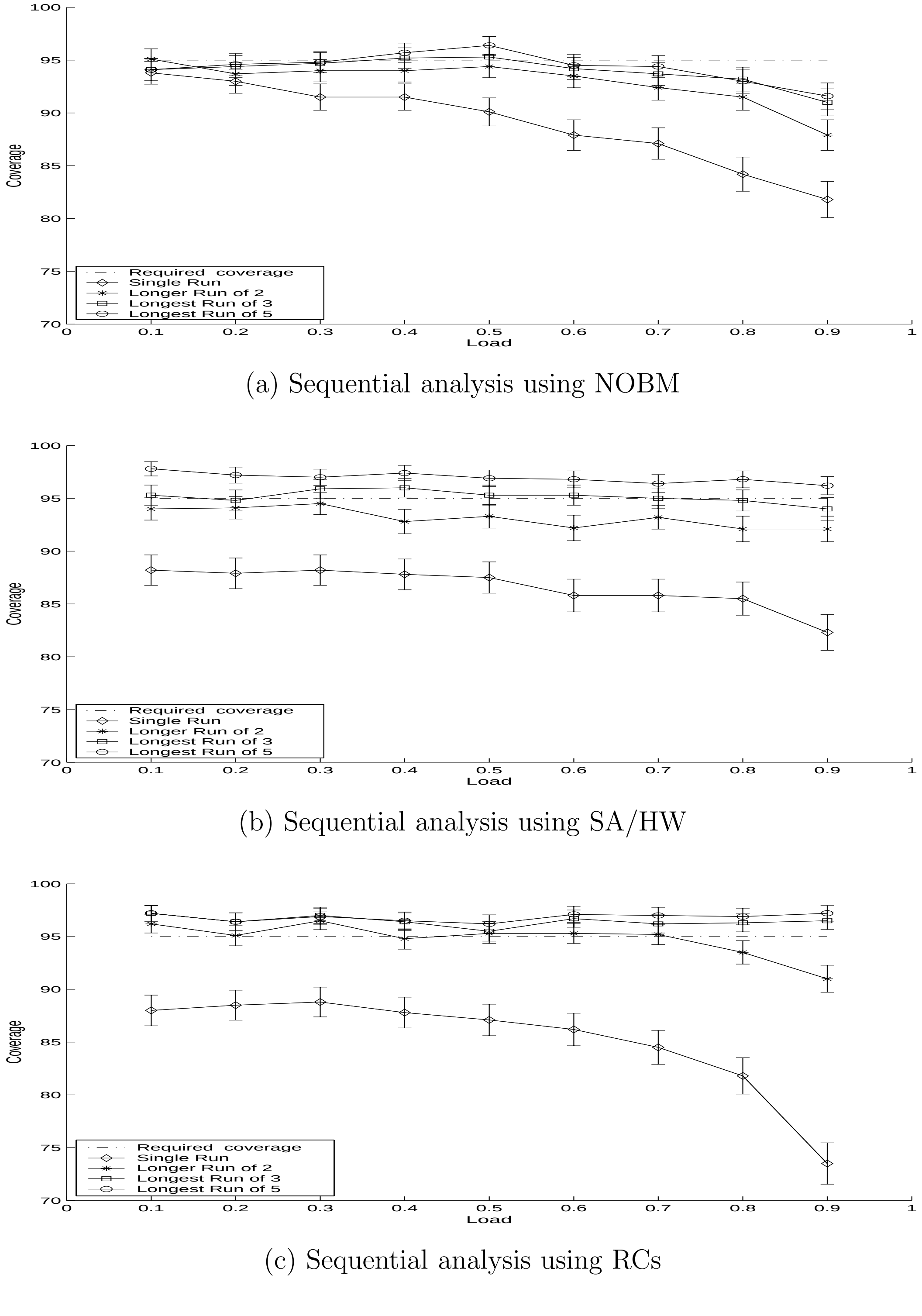


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.6: Coverage of the CIs with Rule I (take the longest of R replications; R = 1, 2, 3 and 5). Estimation of the mean response time in the $M/D/1/\infty$ queueing system



(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.7: Coverage of the CIs with Rule I (take the longest of R replications; R = 1, 2, 3 and 5). Estimation of the mean response time in the $M/H_2/1/\infty$ queueing system

and Figure 3.10 ($M/H_2/1/\infty$). From these, one can see that discarding the results with larger (but still acceptable) levels of the final (reported) relative statistical error *worsens* the coverage, regardless of the number of executed replications, $R > 1$. In fact, a larger R will make the resulting coverage even worse. This is because the simulation producing the most accurate results, in terms of the relative statistical error, has the narrowest CIs. These narrow CIs may sometimes be caused by the sudden (temporary) drop of the required level of relative statistical error, causing accidental stopping with an insufficient number of observations. Consequently, Rule II should not be applied in a practical simulation.

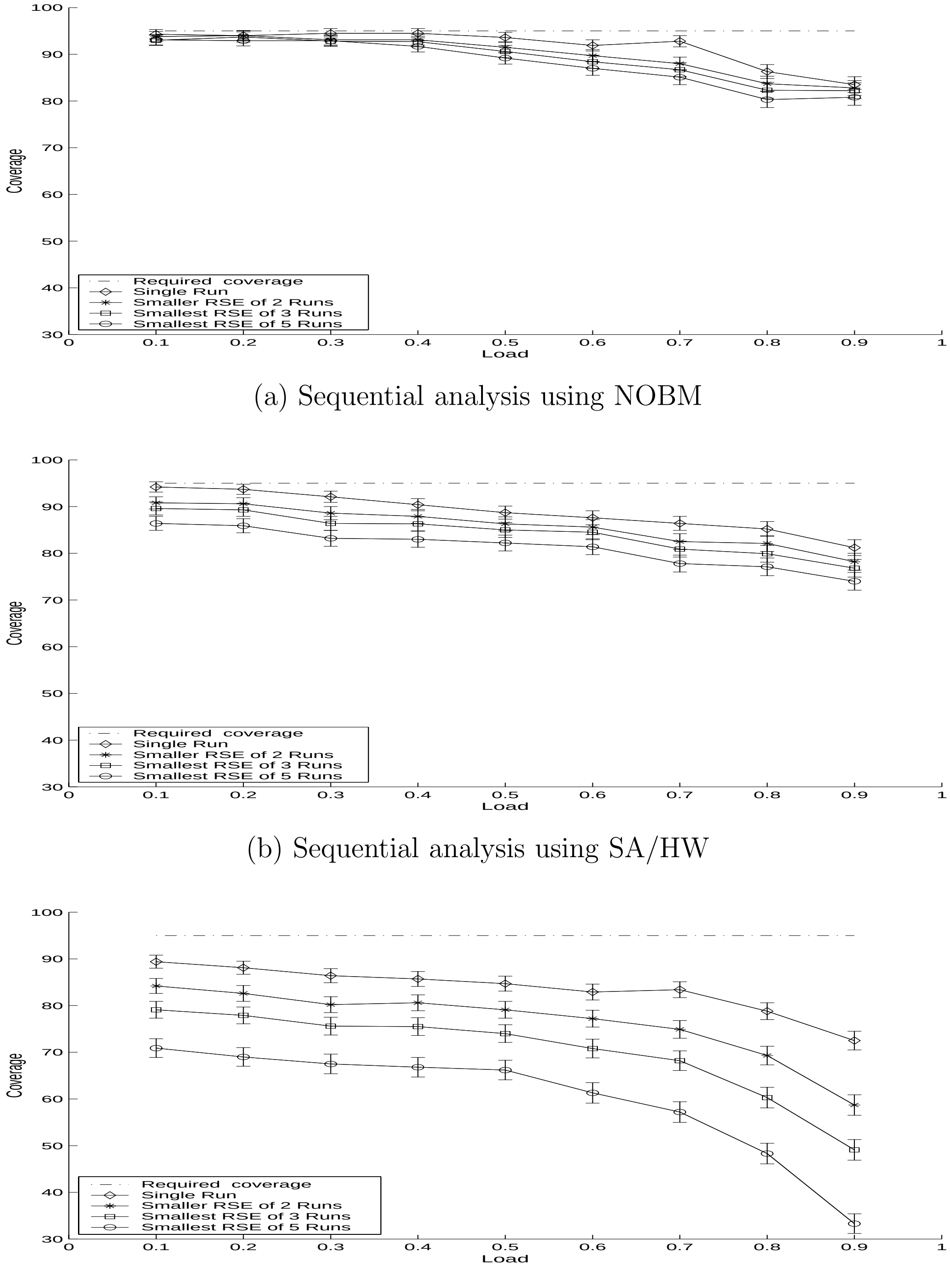
Performance Evaluation: Heuristic Rule III

The application of Rule III, which takes the widest CIs of R replications; $R =$ 1, 2, 3 and 5, is shown in Figure 3.11 ($M/M/1/\infty$), Figure 3.12 ($M/D/1/\infty$), and Figure 3.13 ($M/H_2/1/\infty$). As we can see, taking the simulation results with wider CIs improves the coverage of the final results, regardless of the number of executed replications, where $R > 1$. However, the results of the coverage for each method have not reached the required confidence level of 95%, especially when they are applied in the simulation of heavier loaded queueing systems. Thus, generally speaking, Rule III appears to be unsuitable in a practical simulation.

Performance Evaluation: Heuristic Rule IV

Figure 3.14 ($M/M/1/\infty$), Figure 3.15 ($M/D/1/\infty$), and Figure 3.16 ($M/H_2/1/\infty$), show the effect of applying Rule IV (combining R replications; $R =$ 1, 2, 3 and 5) for each method of simulation output data analysis: NOBM, SA/HW, RCs. The larger the number of replications executed, the better the coverage and also the better (i.e. narrower) the CIs obtained simultaneously.

Generally speaking, there is no need to assume that R is larger than 3 in the case of Rule IV. The reason is that the resulting coverage, obtained by combining $R = 3$ replications, is always between the required level of confidence

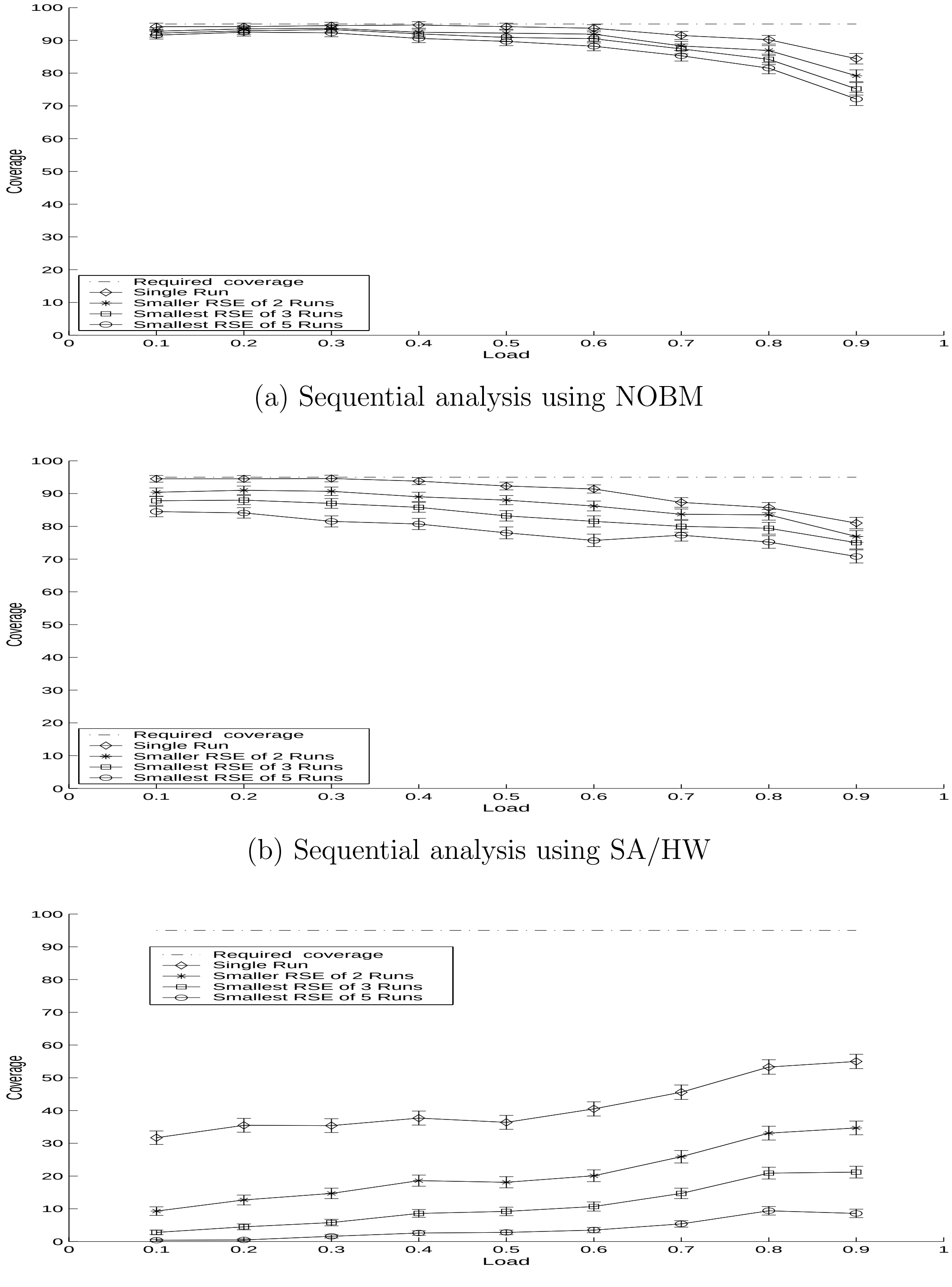


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.8: Coverage of the CIs with Rule II (take the most accurate result out of R results obtained; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/M/1/\infty$ queueing system

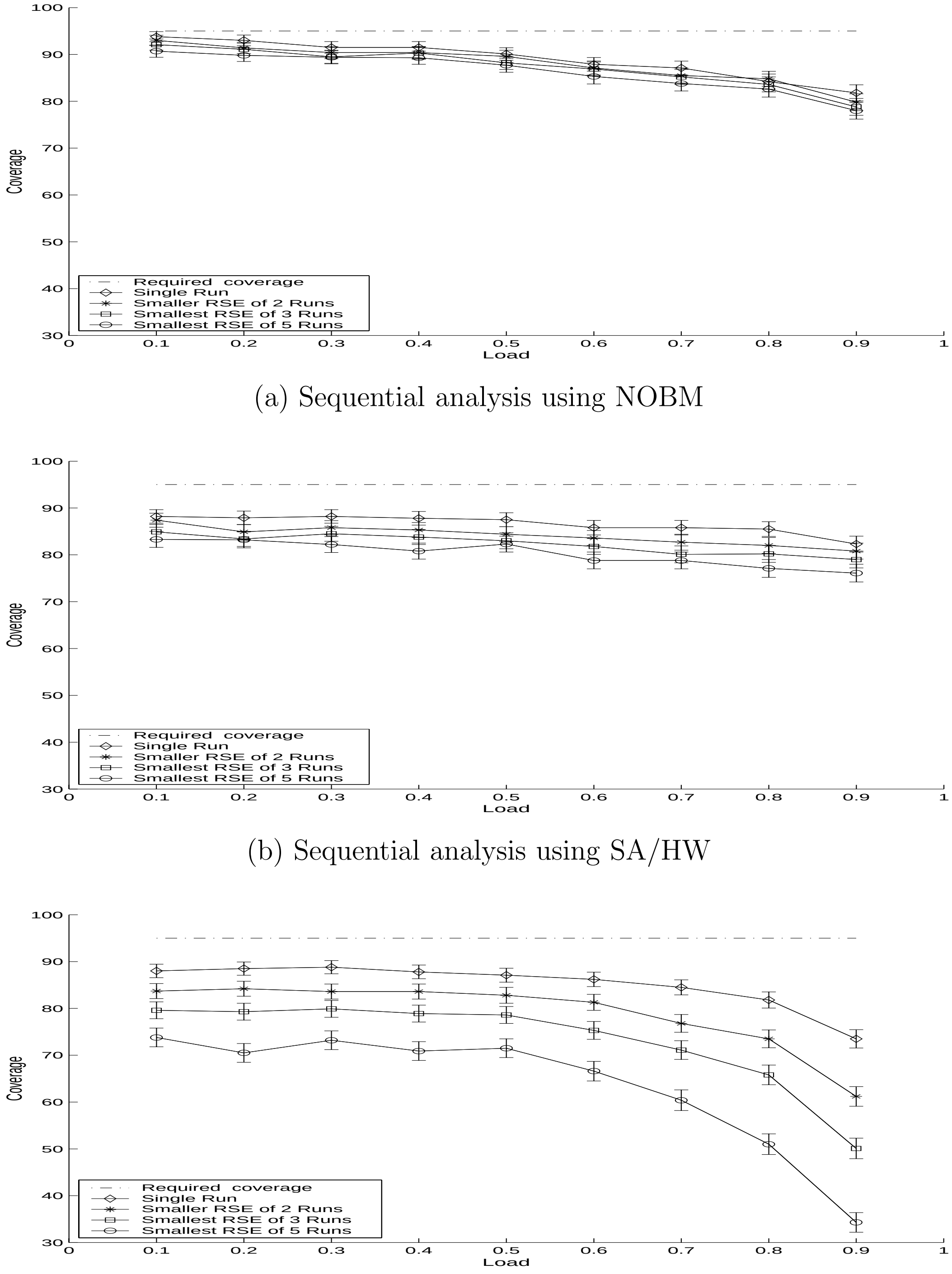


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.9: Coverage of the CIs with Rule II (take the most accurate result out of R results obtained; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/D/1/\infty$ queueing system

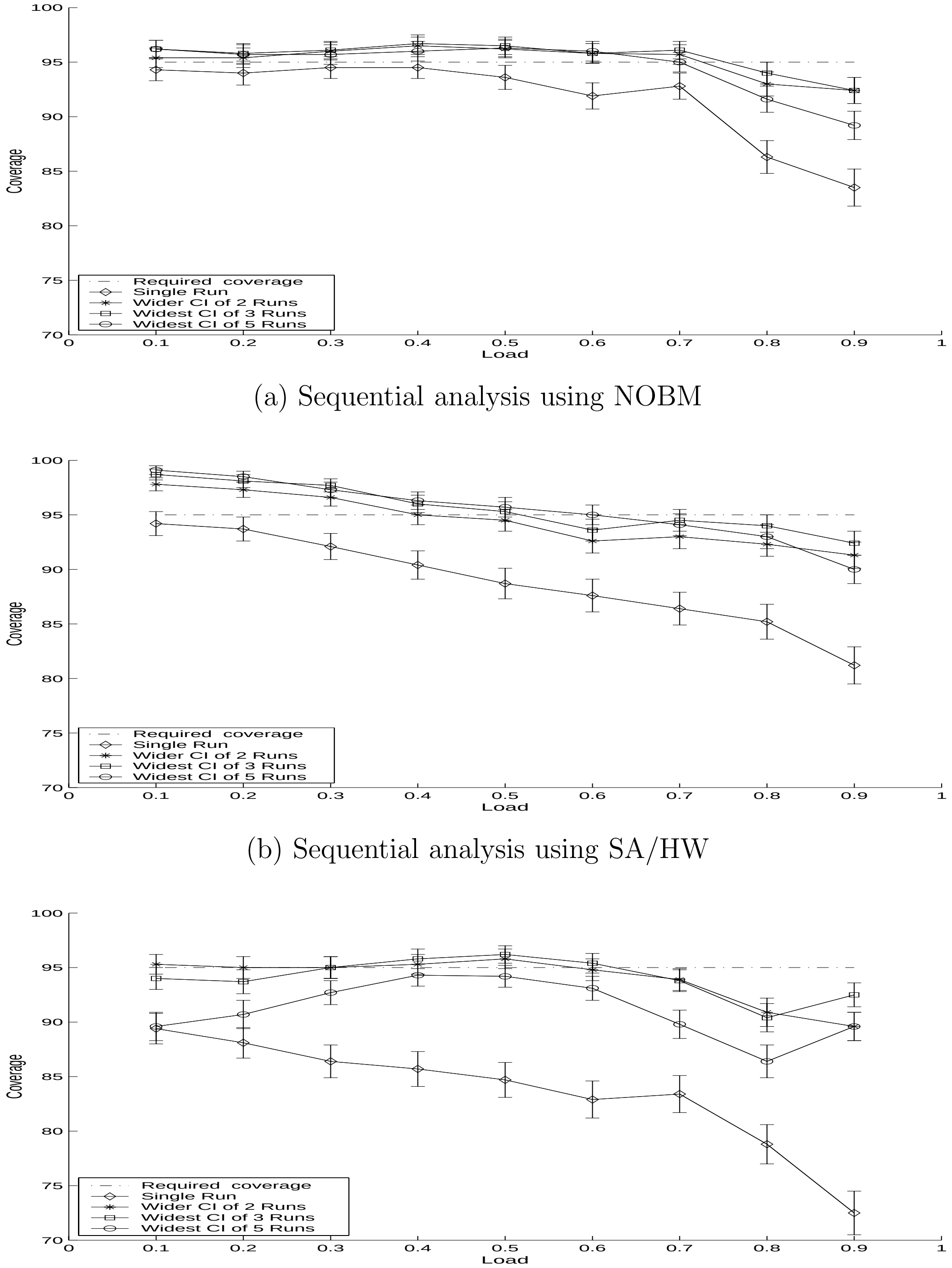


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.10: Coverage of the CIs with Rule II (take the most accurate result out of R results obtained; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/H_2/1/\infty$ queueing system

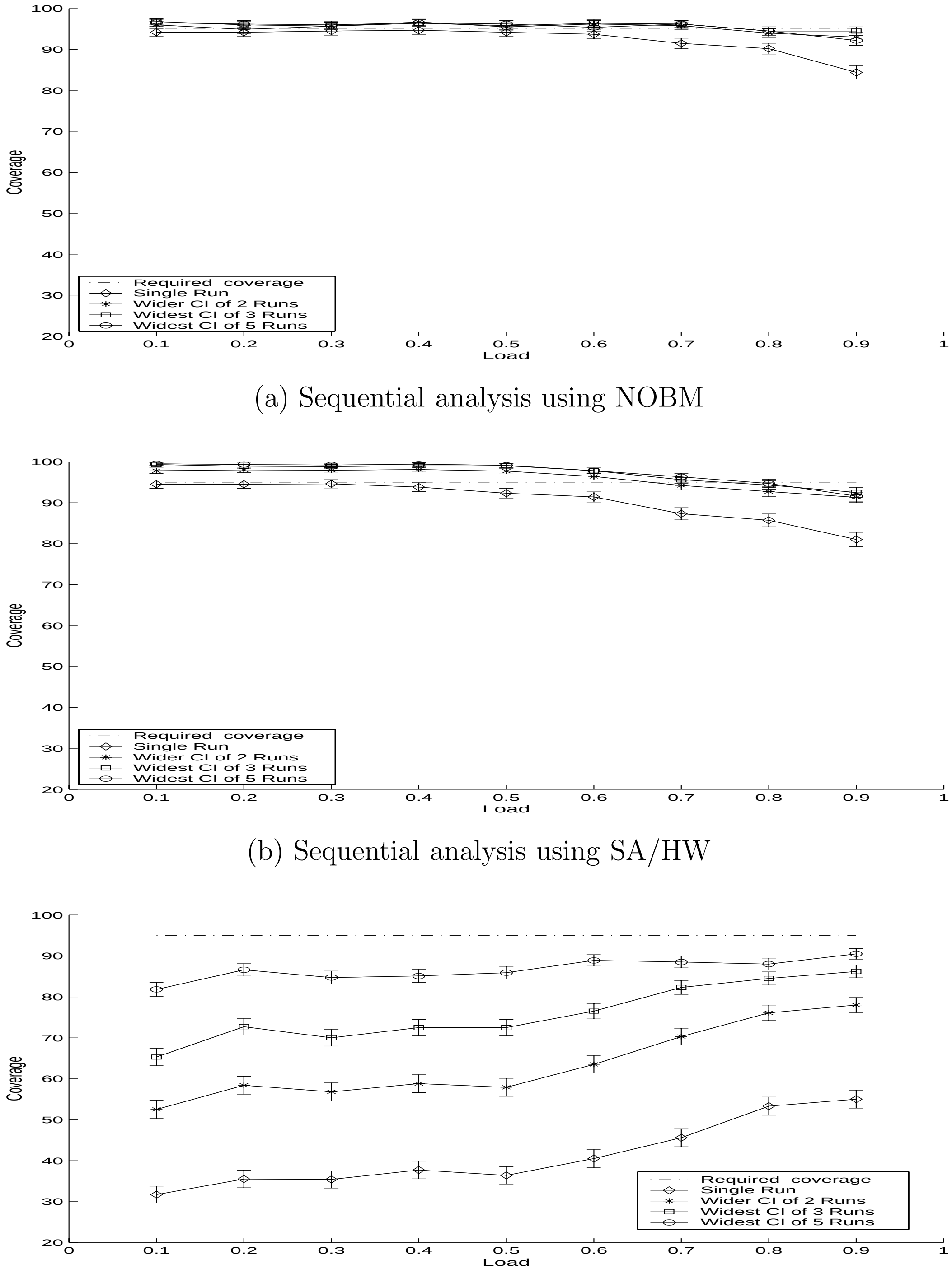


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.11: Coverage of the CIs with Rule III (take the widest CIs of R replications; R = 1, 2, 3 and 5). Estimation of the mean response time in the $M/M/1/\infty$ queueing system

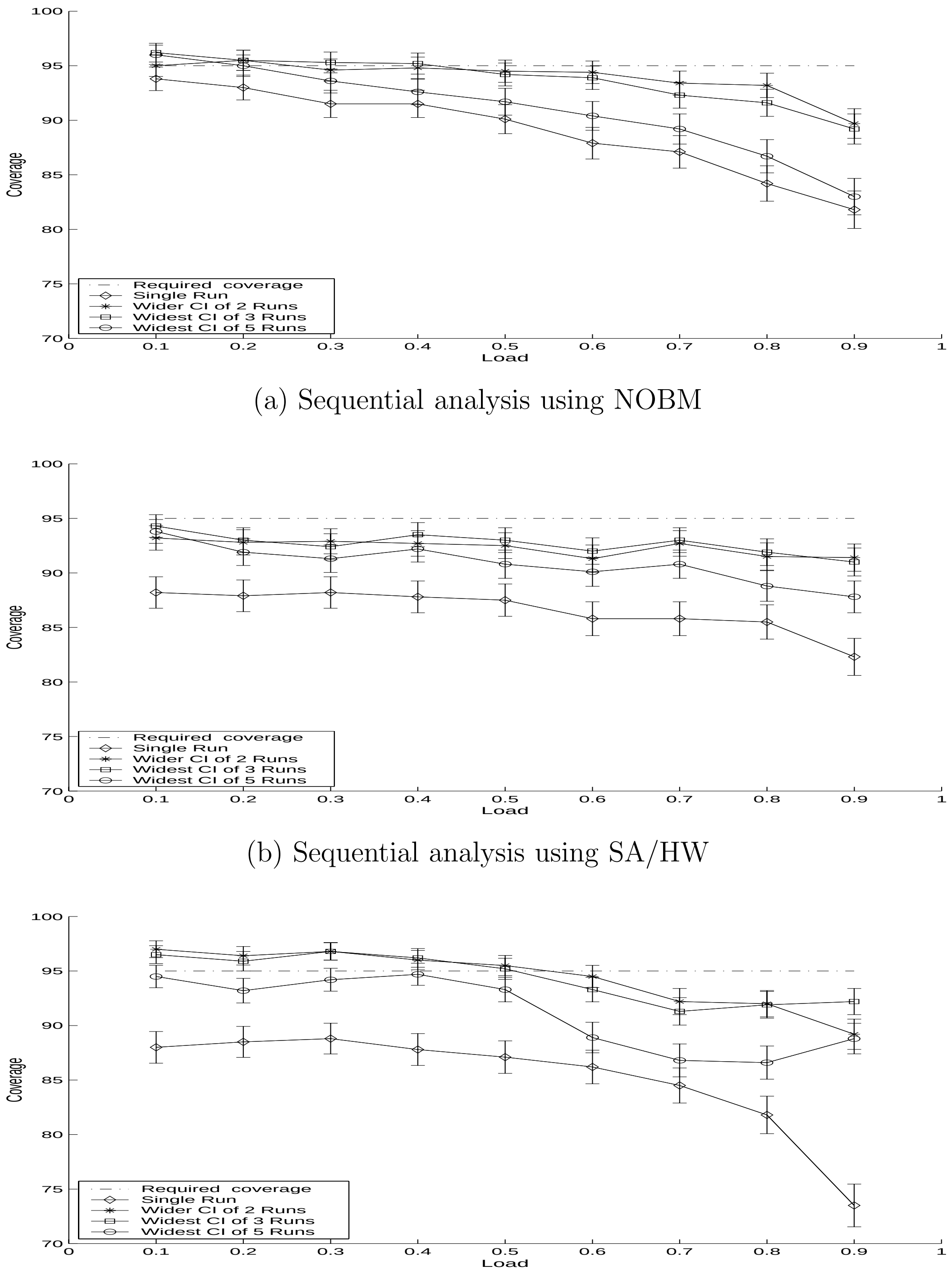


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.12: Coverage of the CIs with Rule III (take the widest CIs of R replications; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/D/1/\infty$ queueing system



(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.13: Coverage of the CIs with Rule III (take the widest CIs of R replications; $R = 1$, 2, 3 and 5). Estimation of the mean response time in the $M/H_2/1/\infty$ queueing system

(0.95) and the maximum level (1.0) for lightly or heavily loaded queueing systems (except the sequential analysis using RCs in the $M/D/1/\infty$ queueing system[3]: see Figure 3.15 (c)). Combining R independent replications together guarantees that final results are produced with a (very) high level of confidence, since the final results are always obtained from a sufficiently large number of observations. Therefore, if one always wants to guarantee the final results having the confidence over the required confidence level in practice, this rule of thumb could be recommended.

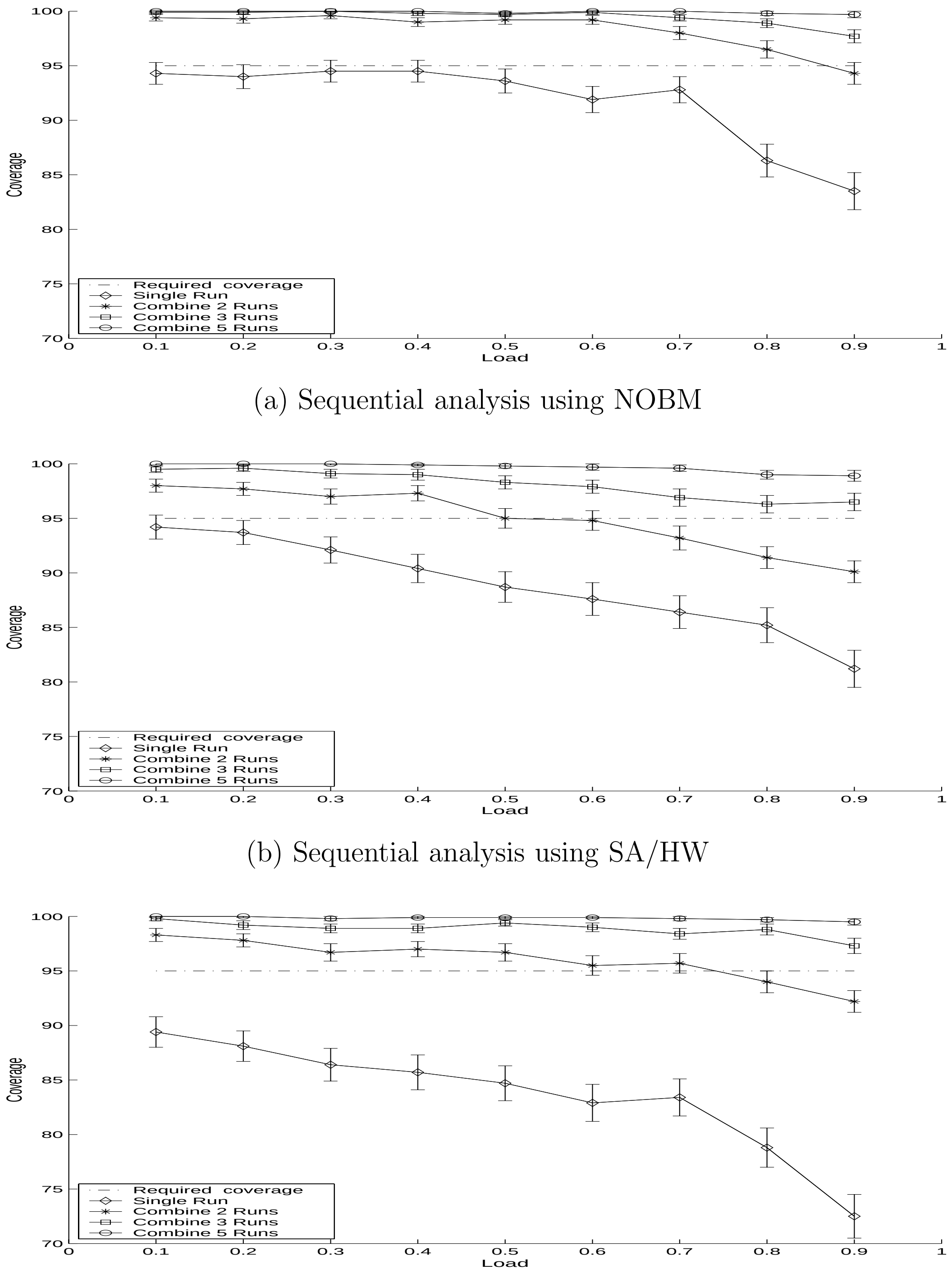
Performance Evaluation: Heuristic Rule V

The results of the coverage when applying Rule V (a combination of Rules III and IV), are depicted in Figure 3.17 ($M/M/1/\infty$), Figure 3.18 ($M/D/1/\infty$), and Figure 3.19 ($M/H_2/1/\infty$). The results are similar to those obtained by applying Rule IV. In general, however, Rule V produces a slightly higher coverage. Therefore, if one always wants to guarantee the final results having the confidence over the required confidence level in practice, this rule of thumb is more desirable than Rule IV.

Comparative Evaluation of Heuristic Rules

In this section, proposed heuristic rules to ensure that the final results of a sequential simulation are not from ‘too short’ simulation runs have been analysed experimentally by applying them to the three different methods of simulation output data analysis: NOBM, SA/HW, RCs, in the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems. The results clearly show that Rules I, IV and V are viable in practice, since they ensure that credible final results are obtained with the required level of confidence or better as the number of replications R increase. The results also show that there is no need to assume R larger than 3.

[3] The unacceptable coverage obtained in the sequential analysis using RCs in the $M/D/1/\infty$ queueing system, even if the combined R is larger than 3, is caused by the very high probability of runs being ‘too short’ (0.732 or less): see Table 3.7.

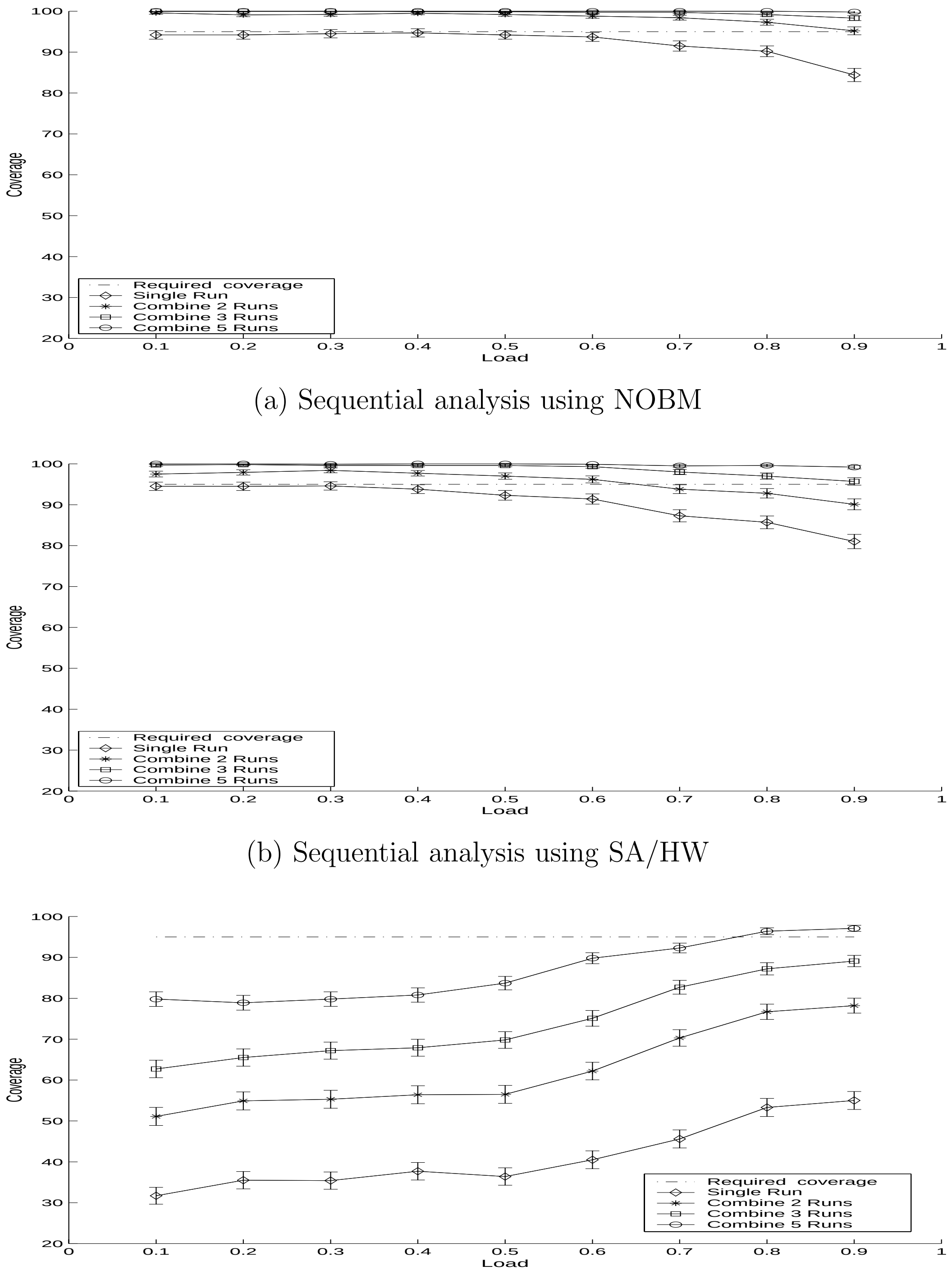


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.14: Coverage of the CIs with Rule IV (combining R replications; R = 1, 2, 3 and 5). Estimation of the mean response time in the $M/M/1/\infty$ queueing system

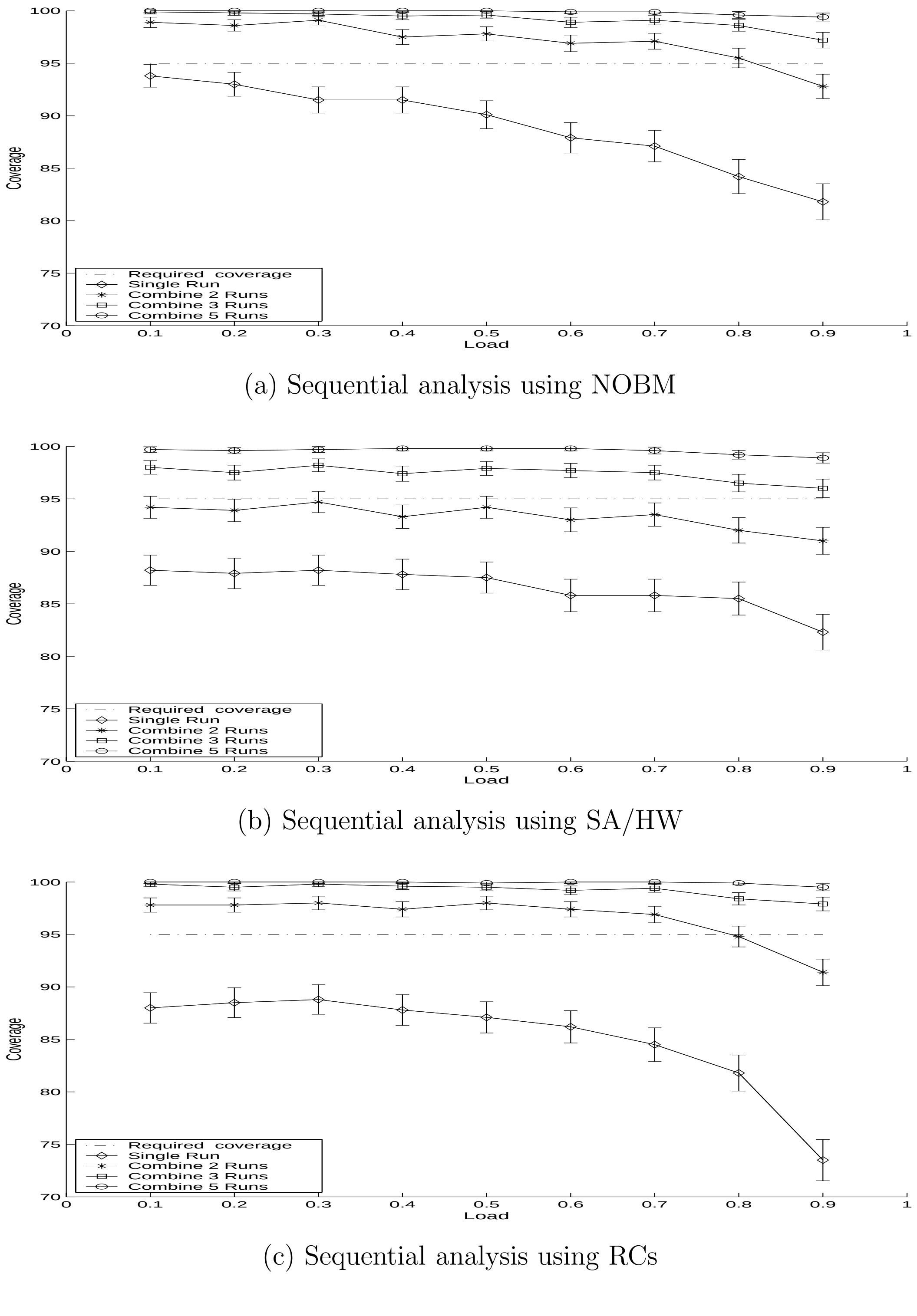


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.15: Coverage of the CIs with Rule IV (combining R replications; R = 1, 2, 3 and 5). Estimation of the mean response time in the $M/D/1/\infty$ queueing system

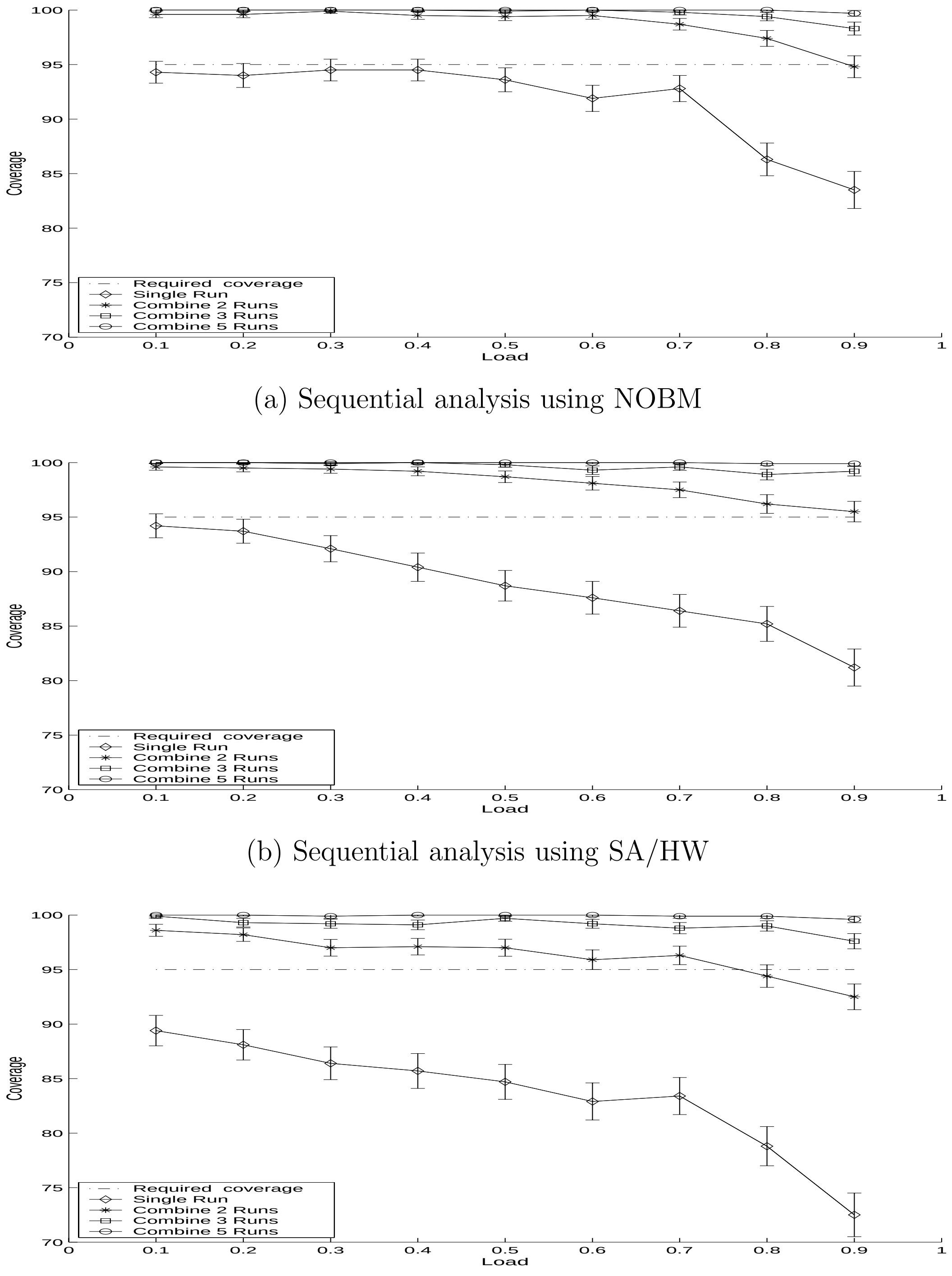


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.16: Coverage of the CIs with Rule IV (combining R replications; R = 1, 2, 3 and 5). Estimation of the mean response time in the $M/H_2/1/\infty$ queueing system

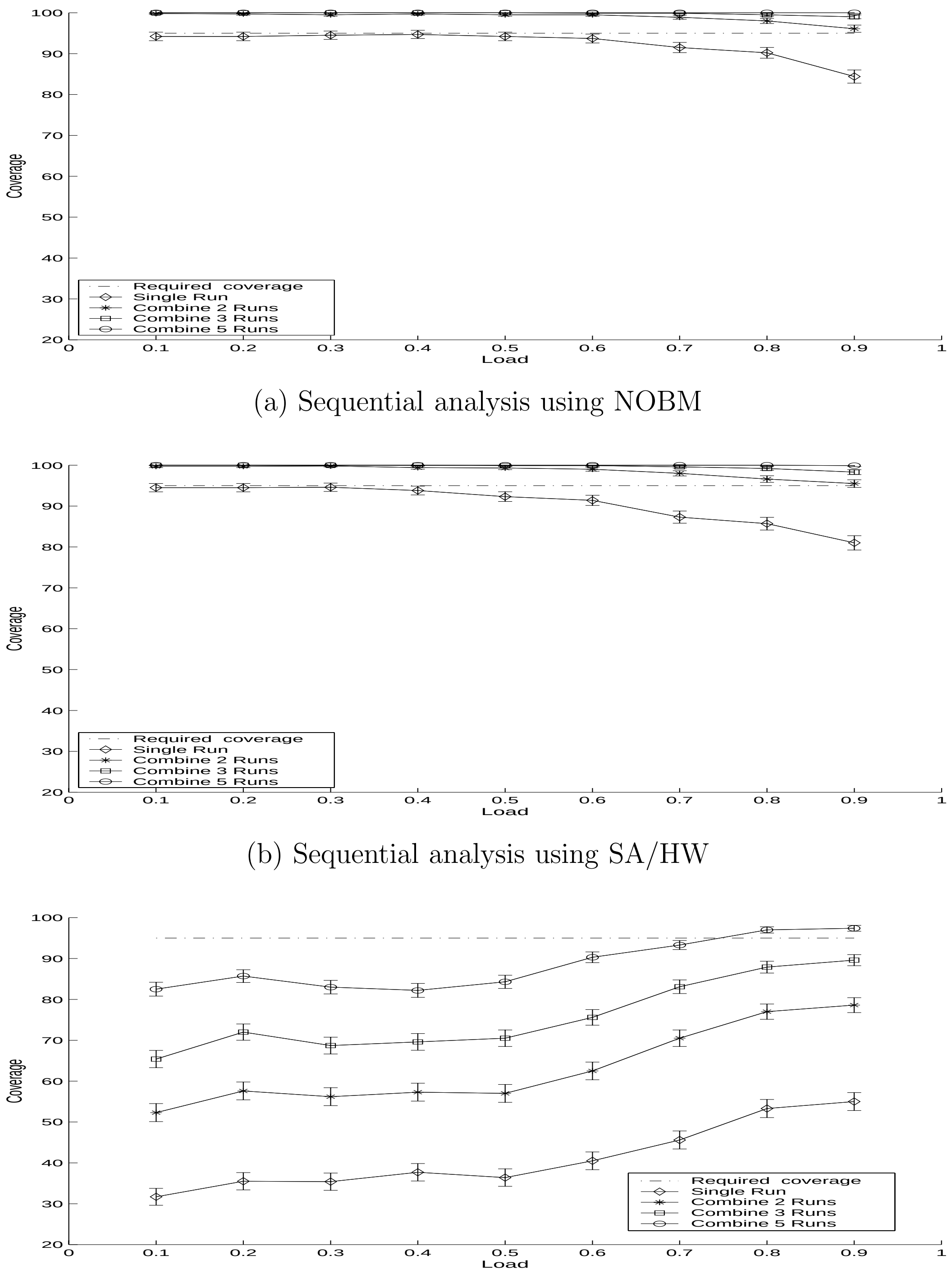


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.17: Coverage of the CIs with Rule V (combination of Rules III and IV). Estimation of the mean response time in the $M/M/1/\infty$ queueing system

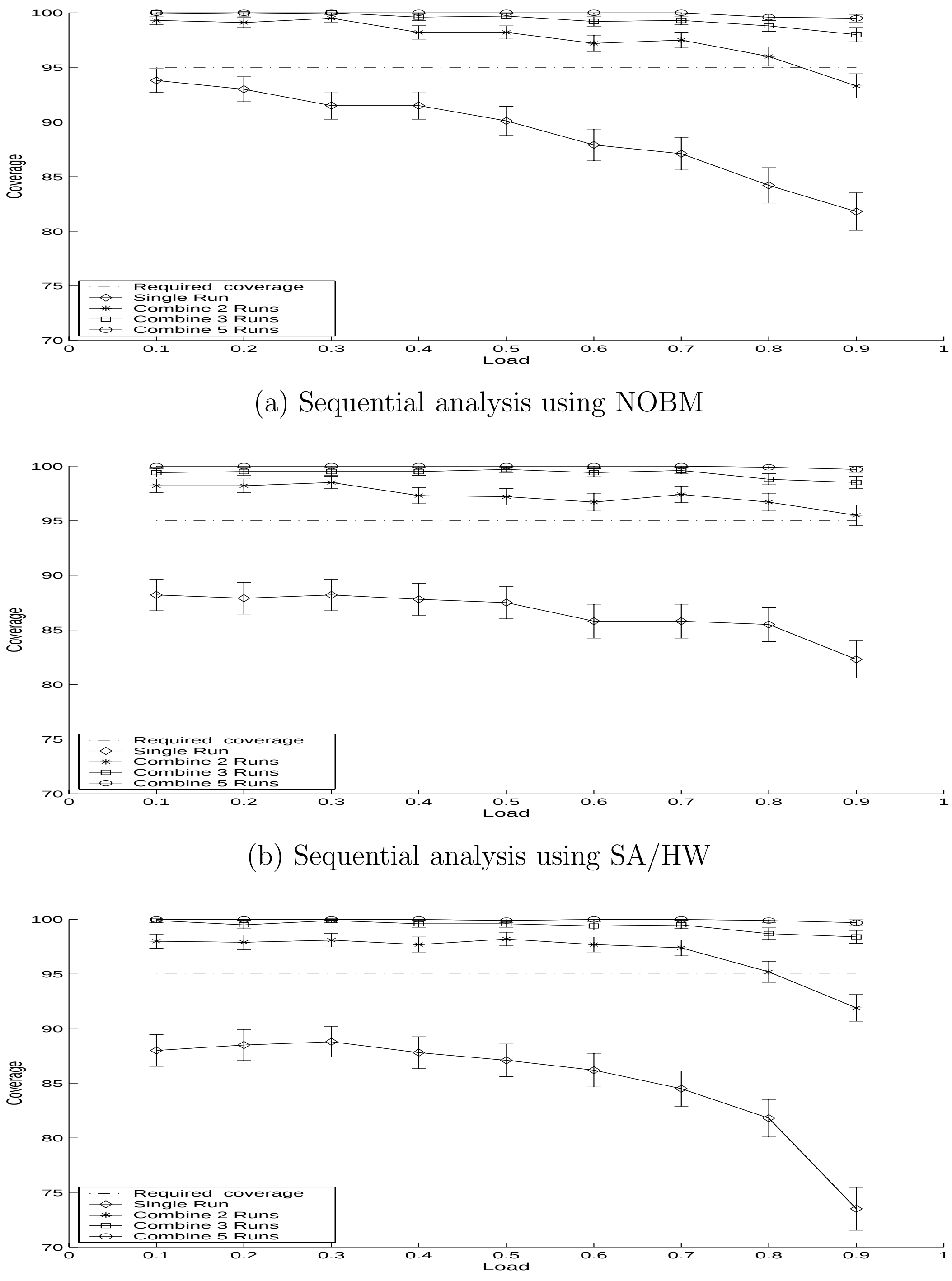


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.18: Coverage of the CIs with Rule V (combination of Rules III and IV). Estimation of the mean response time in the $M/D/1/\infty$ queueing system



(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.19: Coverage of the CIs with Rule V (combination of Rules III and IV). Estimation of the mean response time in the $M/H_2/1/\infty$ queueing system

We have only experimented the proposed heuristic rules with the three different squared coefficient of variation, C^2, for the service times: $C^2 = 0$ (for $M/D/1/\infty$), $C^2 = 1$ (for $M/M/1/\infty$), and $C^2 = 5$ (for $M/H_2/1/\infty$). Therefore, the proposed heuristic rules can only be applicable for simulated processes having the squared coefficient of variation for the service times in similar range. However, one can expect that they could be also used for simulated processes which do not exceed very much the experimented range of the squared coefficients of variation. If one always wants to have the final results within a required level of confidence, Rule I should be the best option. Otherwise, if one wishes to guarantee a high level of confidence, Rules IV and V could be applied. In fact, in the latter case, Rule V is more desirable than Rule IV, since Rule V produces slightly higher coverage.

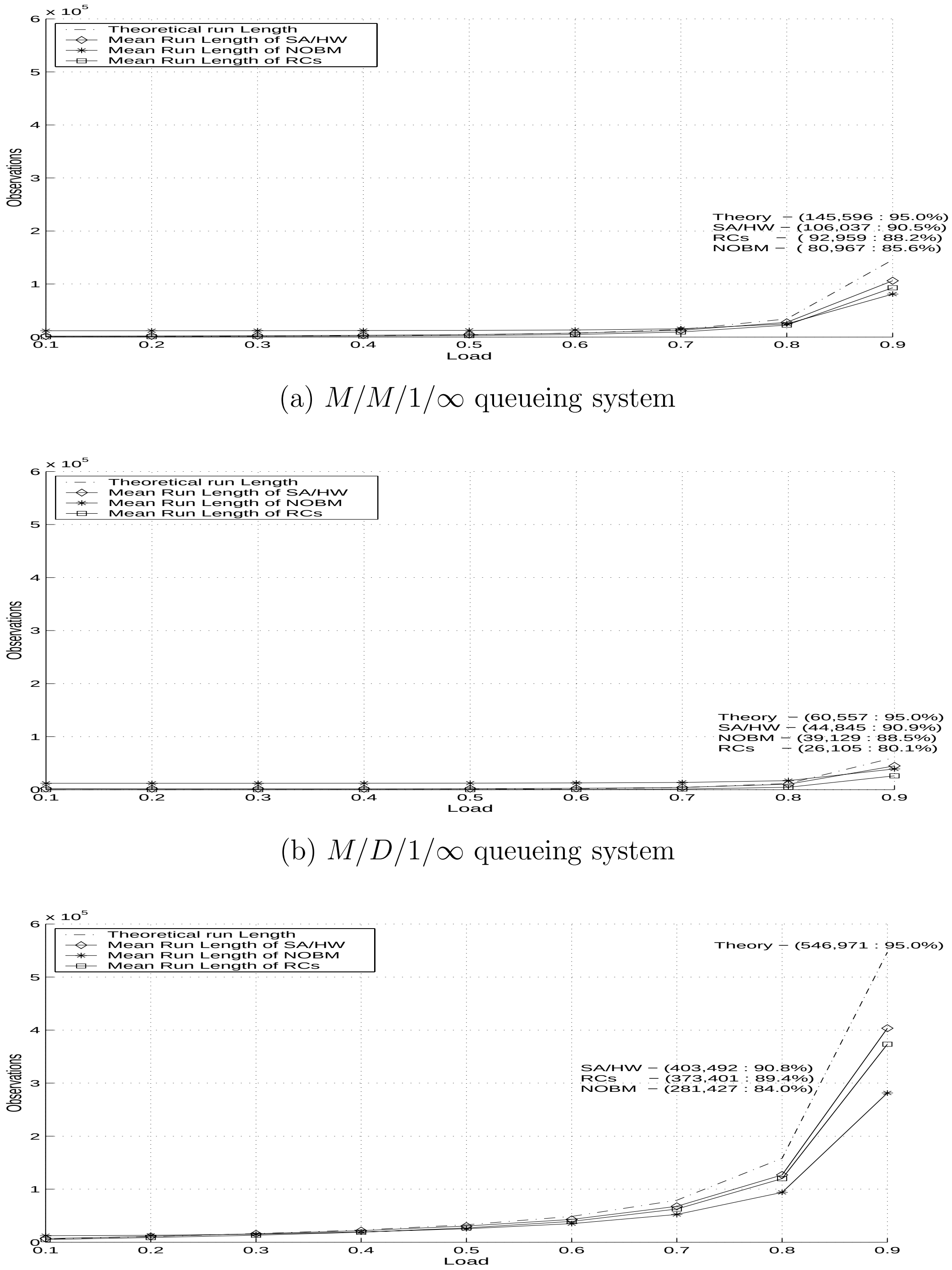
3.5 Theoretical and Experimental Run-Length

The final coverage of the sequential methods of NOBM, SA/HW, and RCs is far from the required level of confidence, especially in highly correlated systems such as the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems; see Figures 2.7 - 2.9 in Chapter 2. This may be because insufficient observations are collected in each simulation run. The average number of observations collected in experiments[4], and the numbers required theoretically[5], when estimating the mean response time for each queueing system, are shown in Figure 3.20. These are obtained from 10,000 independent replications of steady-state simulations, with at least $\epsilon_{max} \cdot 100\% = 10\%$ as the upper level of the acceptable relative statistical error of the final results, at a confidence level of 0.95.

For light traffic intensities, the experimental and theoretical numbers of observations are very close when using RCs and SA/HW for both the $M/M/1/\infty$ and $M/H_2/1/\infty$ queueing systems. However, the finally accepted batch size

[4] Here, in the sequential method of RCs, we present the collected numbers of observations instead of the collected numbers of RCs to facilitate comparisons with NOBM and SA/HW.

[5] Formulae for obtaining the theoretically required run-length of a simulation for stationary queueing systems can be found in Appendix F.



(a) $M/M/1/\infty$ queueing system

(b) $M/D/1/\infty$ queueing system

(c) $M/H_2/1/\infty$ queueing system

Figure 3.20: Mean run-length of 10,000 independent simulation runs of estimating the mean response time

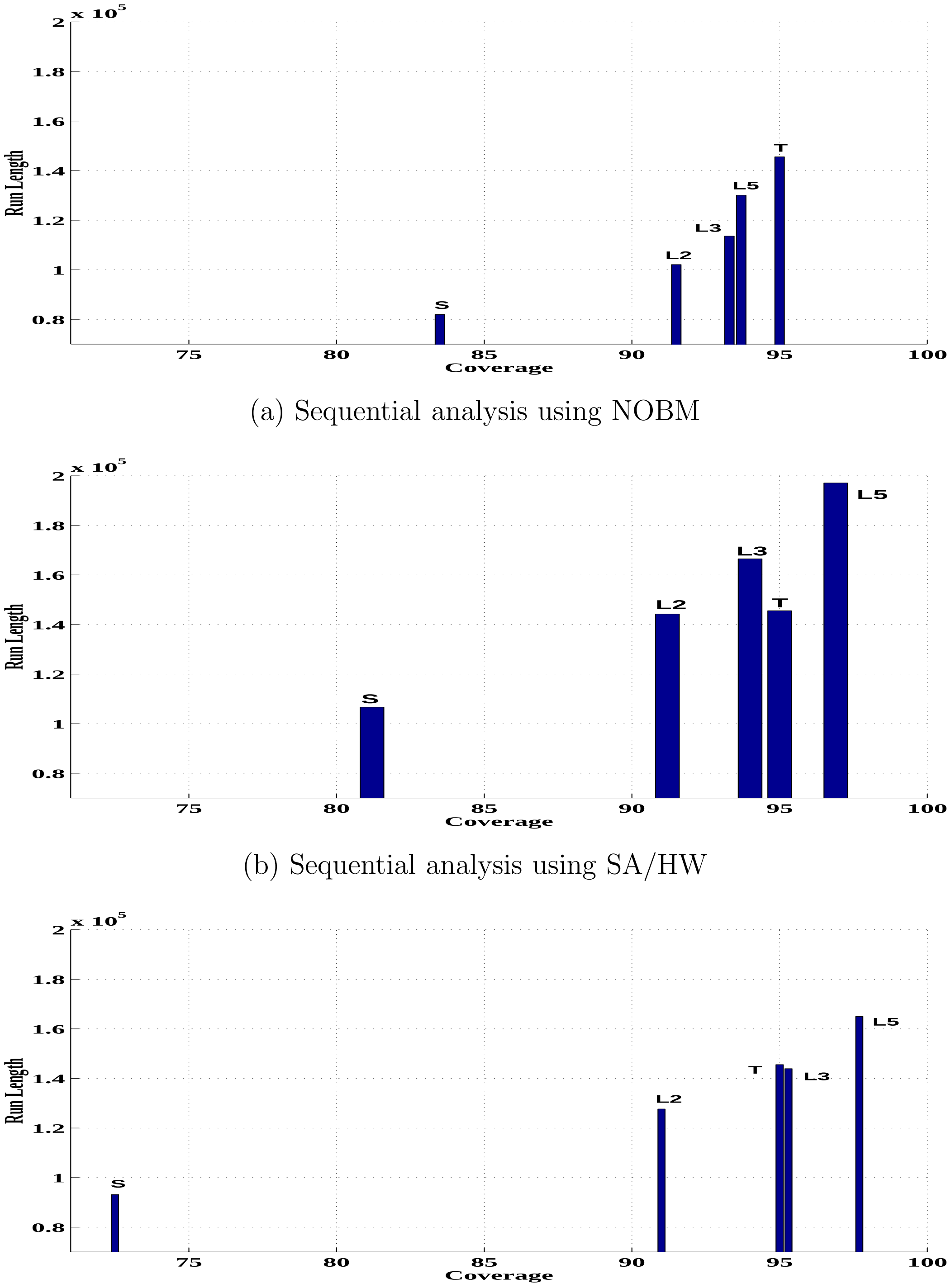
in the NOBM method is large, resulting in some difference between experimental and theoretical numbers. The RCs method seems to produce coverage close to theoretical values at light traffic intensities for both the $M/M/1/\infty$ and $M/H_2/1/\infty$ queueing systems, since it collects the approximate number of observations required in theory. However, at heavier traffic intensities, no method can reach the required CI level of 0.95 because the theoretically required number of observations was not collected.

This definitely indicates that coverage is closely related to the run-lengths of sequential steady-state simulations. That is, one of the reasons causing poor coverage in practical simulations of highly correlated processes, regardless of any simulation output data analysis method used, is that the theoretically required minimum number of observations is not collected.

3.6 Relationship Between Coverage and Run-Length

The relationship between coverage and run-length, when applying Rule I for the sequential NOBM, SA/HW, and RCs (at load 0.9) can be seen in Figure 3.21 ($M/M/1/\infty$), Figure 3.22 ($M/D/1/\infty$), and Figure 3.23 ($M/H_2/1/\infty$). Generally speaking, to ensure that we obtain coverage with an assumed level of confidence, one needs to collect the number of observations closed to that required theoretically, or more, although the relationship between coverage and run-length does depend on the method used in the simulation output data analysis.

Increasing the total number of observations to the number required theoretically seems to be a suitable way of obtaining credible final results. We can never guarantee the assumed exact level of coverage for all simulation models, but we can at least improve the coverage by increasing the number of observations to that required theoretically (if known) or to a sufficiently large number of observations (if the theoretically required number is unknown) by applying Rules I, IV and V.

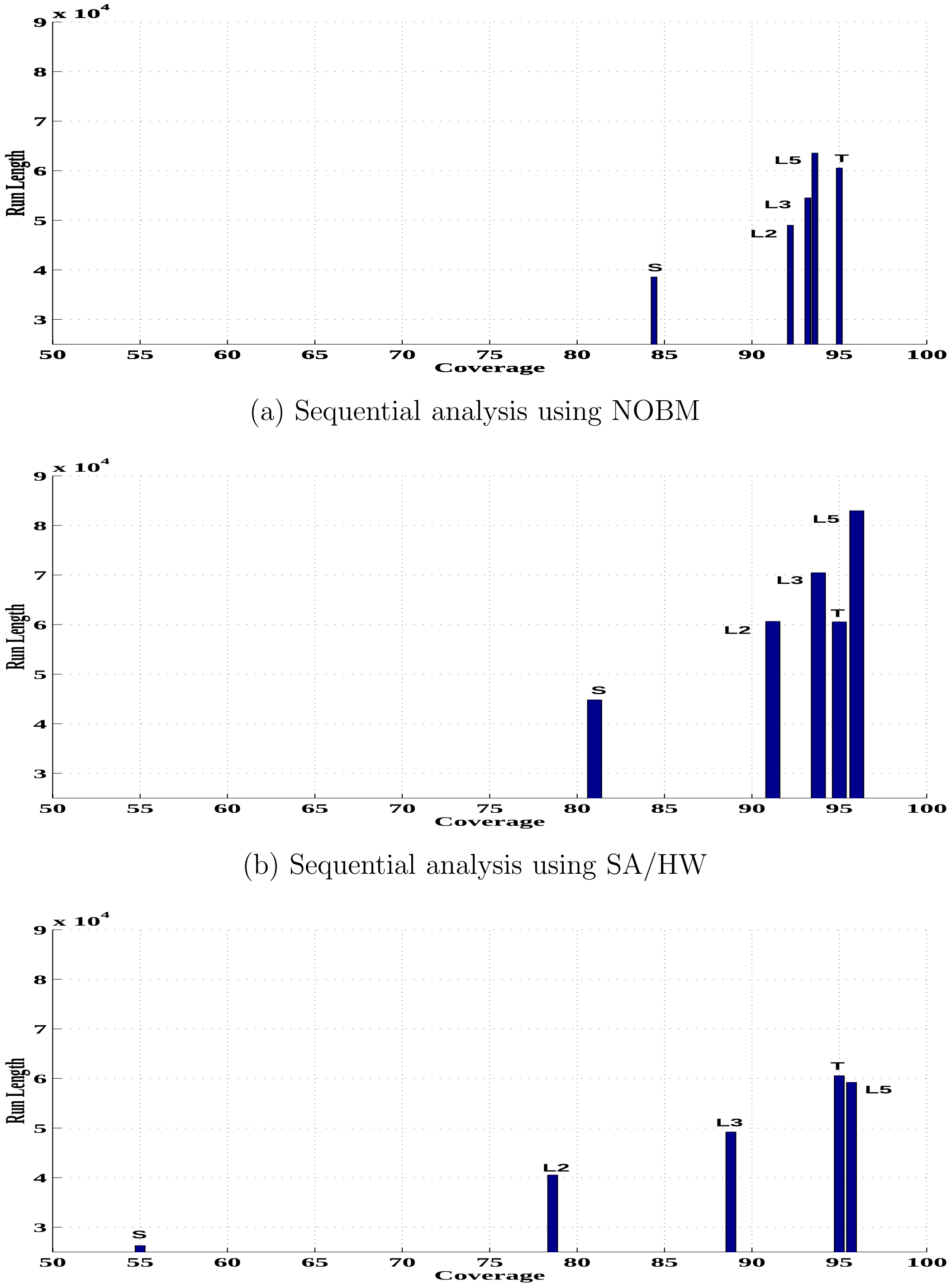


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.21: Coverage over run-lengths with Rule I (at load 0.9, $M/M/1/\infty$). T: theoretical requirement, S: experimental results of a single run, L2: experimental results of a longer run of 2, L3: experimental results of the longest run of 3, and L5: experimental results of the longest run of 5

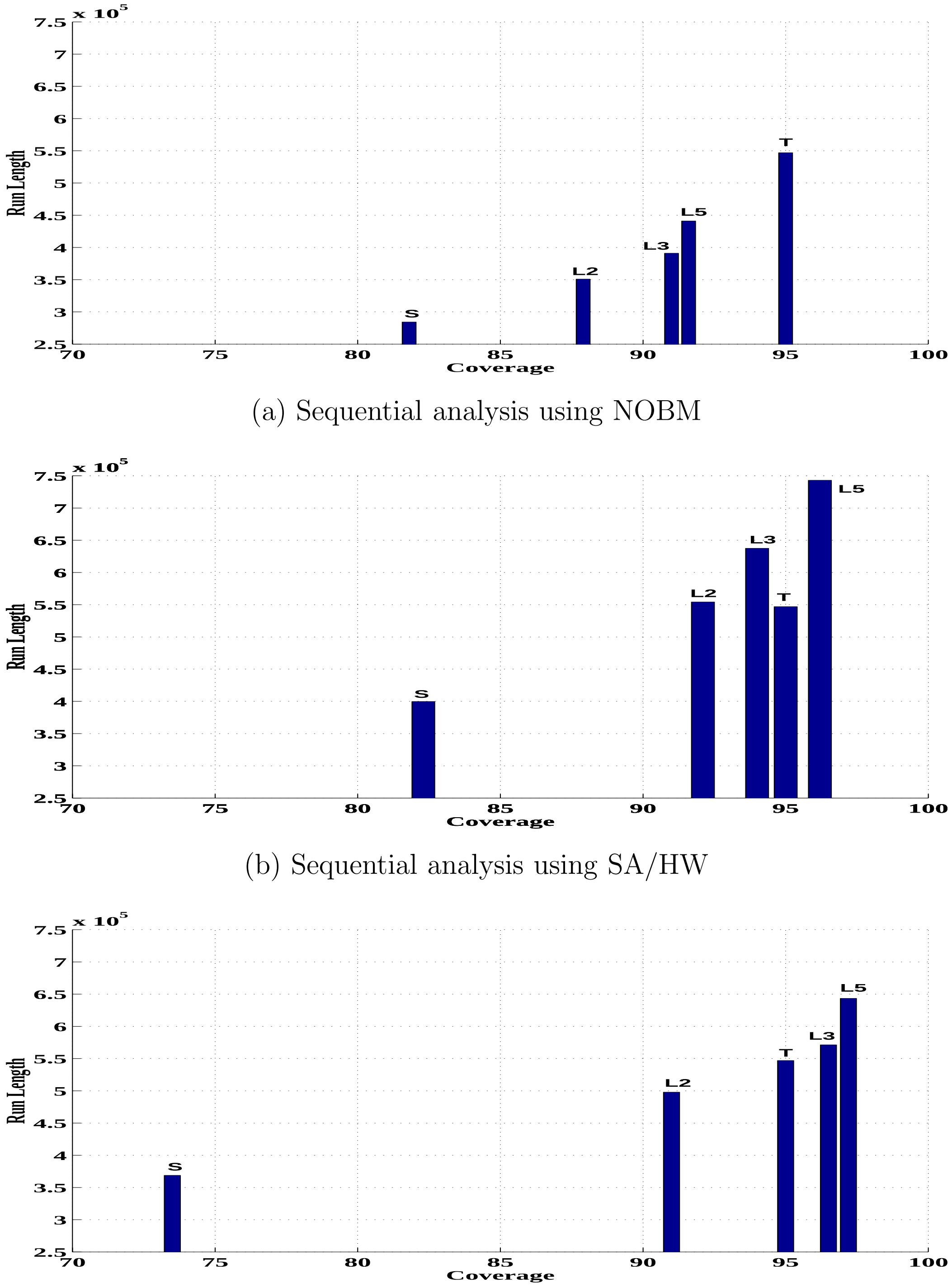


(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.22: Coverage over run-lengths with Rule I (at load 0.9, $M/D/1/\infty$). T: theoretical requirement, S: experimental results of a single run, L2: experimental results of a longer run of 2, L3: experimental results of the longest run of 3, and L5: experimental results of the longest run of 5



(a) Sequential analysis using NOBM

(b) Sequential analysis using SA/HW

(c) Sequential analysis using RCs

Figure 3.23: Coverage over run-lengths with Rule I (at load 0.9, $M/H_2/1/\infty$). T: theoretical requirement, S: experimental results of a single run, L2: experimental results of a longer run of 2, L3: experimental results of the longest run of 3, and L5: experimental results of the longest run of 5

3.7 Conclusions

We have addressed the problem of the statistical correctness of the final simulation results in the context of sequential steady-state simulations, conducted to study long run mean values of performance measures of stable dynamic systems. Typically, in long simulation runs, the convergence of the relative statistical error to its threshold value is very slow but persistent. However, the inherently random nature of output data collected during the stochastic simulation, due to the pseudo-random nature of input data, can cause an accidental, temporary satisfaction of the stopping rule of such a sequential estimation. This is quite frequently associated with producing a 'too short' simulation run having poor coverage. Experimental evidence shows that this phenomenon occurs frequently, with a resulting significant degradation of the coverage of the final results.

We have also proposed five simple heuristic rules of thumb, which are based on two main ideas: (i) using the results from only one run of several executed runs (Rules I to III) or (ii) using the results of all runs without discarding any executed runs (Rules IV and V), that, if applied in practice, can reduce the probability that results come from a prematurely finished simulation run. The effectiveness of these rules is quantitatively assessed using the results of coverage analysis of the three different methods of simulation output data analysis: NOBM, SA/HW, and RCs, in sequential steady-state simulations. Such rules can be easily implemented in simulation packages, offering automated control of the relative statistical error of the final results in a sequential steady-state simulation.

However, no rules can ensure that the final CIs from the sequential stochastic simulation will exactly contain the theoretical value with a probability equal to the assumed confidence level. One of the ongoing problems of research in this area is to find a valid method of analysis (in the sense of coverage) when it is applied to the simulation of highly dynamic stochastic processes, such as heavily loaded queueing systems and telecommunication networks; see for example [106], [134]. At least lowering the probability of using results from 'too short' simulation runs is one of the very few possible practical ways available for simulation practitioners to improve the quality of the final results from their simulation experiments.

Our results show that, to ensure that we obtain the coverage with an assumed level of confidence, one needs to collect the number of observations that is theoretically required, depending on the reference model used in simulations. However, none of the three methods of simulation output data analysis we used collects the theoretically required number of observations in the case of heavily loaded queueing systems. Furthermore, in practice, the theoretically required number of observations is usually unknown. Therefore, we can never guarantee the assumed exact level of coverage with the current state-of-the-art of simulation output data analysis methods in practice, but we can at least improve the coverage by increasing the number of observations to that required theoretically (if known) or to a sufficiently large number of observations (if the theoretically required number of observations is unknown) by applying Rules I, IV and V.

The selection of the appropriate rule depends on the confidence level required. Rule I, which selects the longest run from a few repeated simulation runs, appears to be the most effective in the case where one always wishes to have the final results within an assumed level of confidence, because the coverage from the selected run can be improved to the assumed level of confidence by adjusting the number of replications R. Otherwise, in the case where one always wants to guarantee the final results having a high confidence level, the alternatives are Rules IV and V, as the resulting coverage is always between the assumed level of confidence and the maximum level for lightly or heavily loaded queueing systems. In fact, in the latter case, Rule V is more desirable, since it produces a slightly higher coverage.

Chapter 4

A PROBLEM OF TOO SHORT SEQUENTIAL STEADY-STATE REGENERATIVE SIMULATIONS OF MEAN VALUES

4.1 Introduction

In non-regenerative methods of steady-state simulation output data analysis, such as spectral analysis, batch means, and standardised time series, one should not include data collected during the initial transient period because of the initial non-stationarity. Determination of the end point of the initial transient period is often non-trivial and likely to require sophisticated statistical techniques [16], [57], [180]. Therefore, the method of RCs (regenerative cycles) for simulation output data analysis is a very attractive alternative, because it naturally avoids the problem of the initial transient period. In regenerative stochastic processes, the method of RCs produces batches of random length, which are independent and identically distributed. The final statistical error of the results depends on the number of RCs observed during the entire simulation period. Detailed theoretical discussion and references are documented in Appendix B.3. In this chapter, we investigate the method of RCs to find out how best to tune it for an automated sequential steady-state simulation.

Any stopping criterion for a sequential simulation, for example, the relative statistical error (see Equation (1.2) in Section 1.2), can be used in conjunction with the RC method for estimating steady-state parameters. However, as shown in Chapter 3, sequential steady-state simulation using the three methods of mean value analysis: NOBM, SA/HW, and RCs, can lead to inaccurate results if the experiment stops too early, i.e. when the sequential stopping criterion is accidentally temporarily satisfied. The results presented in Chapter 3 also show that the sequential method of RCs has the most serious problem of early stopping among the three methods.

Lavenberg and Sauer [89] proposed that the simulation should be stopped when a minimum number of RCs are observed (they assumed an arbitrary number of ten RCs as the first checkpoint[1]) and the estimated statistical error reaches the required level. Sauer [156] argued that the simulation run-lengths should be associated with some minimum simulation time. With these approaches, one can run the sequential method of RCs to the minimum run-length of the simulation or for a minimum simulation time. However, the sequential stopping rule even with a minimum number of ten RCs used in [89] as the first checkpoint can not always ensure that a sufficient number of RCs are collected for simulation models having different degrees of autocorrelation. The simulation finished after only collecting the minimum number of RCs can still be '(extremely) too short' if the number of RCs needed to obtain the final results with the assumed level of confidence is very large. As discussed in Chapter 3, this can happen due to the random nature of the fluctuations in the estimated relative statistical error during the stochastic simulation; see,

[1] The first checkpoint at which the relative statistical error $\epsilon(n)$ is computed, can be located after, say, at least two RCs are recorded [89], [165]. Then, the relative statistical error $\epsilon(n)$ can be calculated every k RCs, where $k \geq 1$. The efficiency of computation for checking the stopping rules can be improved by taking larger k.

for example, [130]. Therefore, the sequential stopping rules for the method of RCs should be investigated to find out how '(extremely) too short' simulation runs could be eliminated.

One of the main criteria used to assess the quality of methods of simulation output data analysis in a stochastic simulation is the coverage of the final CIs, defined in Chapter 2. Any good method should produce narrow and stable CIs, and the relative frequency with which such CIs contain the true value of the estimated performance measure should not differ substantially from the assumed theoretical confidence level. In the past, coverage analyses of various sequential stopping rules for the RC method, including those in [89] and [156], were conducted using fixed numbers of replications (for example, 100 and 50 replications, as [89] and [156], respectively). However, as shown in Chapter 2 (see for example Figure 2.5), such a fixed number of replications for coverage analysis is difficult to predict. Therefore, to secure statistically accurate final results, coverage analysis for the sequential methods of RCs should be conducted following the sequential rules discussed in Chapter 2.

In Section 4.2, we summarise the four selected ratio estimators of the mean used in sequential version of the RCs method: the classical estimator, the Beale estimator, the jackknife estimator, and the Tin estimator. We document a problem of early stopping in the sequential method of RCs and a solution, based on experimental results, in Section 4.3 and Section 4.4, respectively. The numerical results of the coverage analysis of the sequential method of RCs with a proposed solution applied for estimating steady-state means are reported in Section 4.5.

4.2 Ratio Estimators for Use in the Sequential Method of RCs

The RC method[2] usually uses the ratio of two means to estimate steady-state parameters. Choice of the regenerative state used for making batches

[2] Detailed discussion of the RC method for simulation output data analysis can be found in Appendix B.3. Notations and definitions used in this section follow Appendix B.3.

of random length from the collected observations is an important parameter. With this method, the initialisation bias is eliminated, but new sources of systematic errors caused by the use of estimators in the form of ratios are introduced [12]. Several estimators have been proposed to reduce these errors [15], [110], [117]. We have selected and summarised only four estimators: the classical estimator, the Beale estimator, the jackknife estimator, and the Tin estimator, since these can be easily implemented in sequential steady-state simulations.

Classical Estimator

The simplest ratio estimator, known as the *classical estimator*; see Appendix B.3, of the steady-state mean for the RC method based on n RCs is given by

$$\hat{r}(n) = \frac{\overline{y}(n)}{\overline{a}(n)}, \tag{4.1}$$

where $\overline{y}(n)$ is the mean of y_i (where $1 \leq i \leq n$) which is the sum of observations in the i-th RC, for example, the sum of the waiting times in the i-th RC, and $\overline{a}(n)$ is the mean of a_i (where $1 \leq i \leq n$) which is the number of observations in the i-th RC.

Following the central limit theorem:

$$\frac{\sqrt{n}\{\hat{r}(n) - \mu\}}{s(n)/\overline{a}(n)} \rightarrow N(0,1), \tag{4.2}$$

where μ is the steady-state mean, $s(n)$ is the point estimate for σ based on n RCs, and $N(0,1)$ is the normal distribution with mean 0 and standard deviation 1, obtained with a probability of one as $n \rightarrow \infty$ [22], [165]. A $100(1-\alpha)\%$ CI for the steady-state mean obtained with the classical estimator is given by

$$\hat{r}(n) \pm \frac{s(n)t_{n-1,1-\alpha/2}}{\overline{a}(n)\sqrt{n}}, \tag{4.3}$$

where $t_{n-1,1-\alpha/2}$, for $0 < \alpha < 1$, is the upper $(1-\alpha/2)$ critical point from the Student t-distribution with $n-1$ degrees of freedom [21], [71], [165].

Beale Estimator

A point estimator, known as the *Beale estimator*, has been shown to reduce the bias[3] of the classical estimator of Equation (4.1) [71]. Using the Beale estimator, the point estimate for the steady-state mean for the RC method based on n RCs is given by

$$\hat{r}_b(n) = \frac{\overline{y}(n)}{\overline{a}(n)} \cdot \frac{(1 + \frac{s_{12}^2(n)}{n\overline{y}(n)\overline{a}(n)})}{(1 + \frac{s_{22}^2(n)}{n\overline{a}(n)\overline{a}(n)})}, \tag{4.4}$$

where $\overline{y}(n)$ is the mean of y_i (where $1 \leq i \leq n$) which is the sum of observations in the i-th RC, for example, the sum of the waiting times in the i-th RC, and $\overline{a}(n)$ is the mean of a_i (where $1 \leq i \leq n$) which is the number of observations in the i-th RC. $s_{12}^2(n)$ is the estimate of covariance for $\overline{y}(n)$ and $\overline{a}(n)$, and $s_{22}^2(n)$ is the estimate of variance for $\overline{a}(n)$. The Beale estimator reduces the bias of the classical estimator from $O(1/n)$ to $O(1/n^2)$ [71], [165].

Since $\sqrt{n}\{\hat{r}(n) - \hat{r}_b(n)\} \rightarrow 0$ as $n \rightarrow \infty$ with a probability of one, one can replace $\hat{r}(n)$ in Equation (4.2) by $\hat{r}_b(n)$ [22]. Then, a $100(1-\alpha)\%$ CI for the steady-state mean obtained with the Beale estimator is given by

$$\hat{r}_b(n) \pm \frac{s(n)t_{n-1,1-\alpha/2}}{\overline{a}(n)\sqrt{n}}, \tag{4.5}$$

where $s(n)$ is the point estimate for σ based on n RCs, and $t_{n-1,1-\alpha/2}$, for $0 < \alpha < 1$, is the upper $(1-\alpha/2)$ critical point from the Student t-distribution with $n-1$ degrees of freedom [22], [71], [165].

[3] In general, the expectation of a ratio is not equal to the ratio of the expectations for any finite n RCs [12]. As a consequence of the strong law of large numbers; i.e., $n \rightarrow \infty$,

$$E\left[(\frac{1}{n}\sum_{i=1}^{n} y_i)/(\frac{1}{n}\sum_{i=1}^{n} a_i)\right] = E[y_i]/E[a_i]$$

with a probability of one, where y_i is the sum of the parameter of interest in the i-th RC and a_i is the length of the i-th RC. However, for any finite n,

$$E\left[(\frac{1}{n}\sum_{i=1}^{n} y_i)/(\frac{1}{n}\sum_{i=1}^{n} a_i)\right] \neq E[y_i]/E[a_i],$$

except in trivial cases [12], [165].

Jackknife Estimator

A version of the *jackknife estimator* was constructed by Miller; see [71]. Using the jackknife estimator, the point estimate for the steady-state mean for the RC method based on n RCs is given by

$$\hat{r}_j(n) = \frac{1}{n}\sum_{i=1}^{n}\theta_i, \tag{4.6}$$

where $\theta_i = n(\overline{y}/\overline{a}) - (n-1)\left(\sum_{k=1,k\neq i}^{n} y_k / \sum_{k=1,k\neq i}^{n} a_k\right)$ for $i = 1, 2, ..., n$. Here, y_k is the sum of observations in the k-th RC and a_k is the number of observations in the k-th RC. The jackknife estimator also reduces the bias of the classical estimator from $O(1/n)$ to $O(1/n^2)$ [71], [93], [165].

Let

$$s_j^2(n) = \frac{\sum_{i=1}^{n}\{\theta_i - \hat{r}_j(n)\}^2}{n-1} \tag{4.7}$$

be the estimator of variance $\sigma^2(n)$ for the jackknife estimator. Then the following limit result provides a basis for a CI of the jackknife estimator:

$$\frac{\sqrt{n}\{\hat{r}_j(n) - \mu\}}{s_j(n)} \rightarrow N(0,1), \tag{4.8}$$

as $n \rightarrow \infty$ with a probability of one [22], [93]. Therefore, a $100(1-\alpha)\%$ CI for the steady-state mean μ obtained with the jackknife estimator is given by

$$\hat{r}_j(n) \pm \frac{s_j(n)t_{n-1,1-\alpha/2}}{\sqrt{n}}, \tag{4.9}$$

where $t_{n-1,1-\alpha/2}$, for $0 < \alpha < 1$, is the upper $(1-\alpha/2)$ critical point from the Student t-distribution with $n-1$ degrees of freedom [22], [93], [165].

Tin Estimator

A point estimator, known as the *Tin estimator*, has been proposed by Tin; see [71]. The point estimate using the Tin estimator for the steady-state mean for the RC method based on n RCs is given by

$$\hat{r}_t(n) = \frac{\overline{y}(n)}{\overline{a}(n)} \cdot \left[1 + \frac{1}{n}\left(\frac{s_{12}^2(n)}{\overline{y}(n)\overline{a}(n)} - \frac{s_{22}^2(n)}{\overline{a}(n)\overline{a}(n)}\right)\right], \tag{4.10}$$

where $\overline{y}(n)$ is the mean of y_i (where $1 \leq i \leq n$) which is the sum of observations in the i-th RC, for example, the sum of the waiting times in the i-th RC, and $\overline{a}(n)$ is the mean of a_i (where $1 \leq i \leq n$) which is the number of observations in the i-th RC. $s_{12}^2(n)$ is the estimate of covariance for $\overline{y}(n)$ and $\overline{a}(n)$, and $s_{22}^2(n)$ is the estimate of variance for $\overline{a}(n)$. The Tin estimator also reduces the bias of the classical estimator from $O(1/n)$ to $O(1/n^2)$ [71], [165].

Since $\sqrt{n}\{\hat{r}(n) - \hat{r}_t(n)\} \rightarrow 0$ as $n \rightarrow \infty$ with a probability of one as with the Beale estimator, one can also replace $\hat{r}(n)$ in Equation (4.2) by $\hat{r}_t(n)$ [22]. Then, a $100(1-\alpha)\%$ CI for the steady-state mean is given by

$$\hat{r}_t(n) \pm \frac{s(n)t_{n-1,1-\alpha/2}}{\overline{a}(n)\sqrt{n}}, \tag{4.11}$$

where $s(n)$ is the point estimate for σ based on n RCs, and $t_{n-1,1-\alpha/2}$, for $0 < \alpha < 1$, is the upper $(1-\alpha/2)$ critical point from the Student t-distribution with $n-1$ degrees of freedom [22], [71], [165].

Comments

Several further alternative ratio estimators for reducing the bias of the classical estimator for the RC method have been proposed in [15], [61], [117]. Comparative studies of some ratio estimators were conducted by D. L. Iglehart [71]. The results presented in [71] show that the jackknife estimator is better than the classical estimator, particularly, for short simulation runs; also see [93]. As the length of the simulation increases, however, the jackknife estimator produces similar results. In particular, it requires twice as much memory (for saving the entire sequence of y_i and a_i) and longer time, and slightly more complex programming than the classical estimator [71]. This means that for long simulation runs such as sequential steady-state simulations, the jackknife estimator has no benefit, since it requires double the memory requirement of the classical estimator without any significant improvement.

The Beale and Tin estimators produce less biased estimates than the classical estimator [71]. Nevertheless, the classical estimator is the most recommended one for interval estimates [71]. It is also the easiest to program and produces quite good results, in terms of coverage, for long simulation runs without any extra cost in memory [71]. Therefore, the classical estimator remains an attractive candidate for long simulation runs, especially when estimating CIs in sequential steady-state simulations.

4.3 Sequential Method of RCs: A Problem of Early Stopping

The stopping rule based on the relative statistical error of Equation (1.2) in Section 1.2 should be modified for a sequential method of RCs[4], based on n RCs, as follows:

$$\epsilon(n) = \frac{\Delta(n)}{\hat{r}(n)}, \tag{4.12}$$

where $\hat{r}(n)$ is the classical point estimator given in Equation (4.1), $\Delta(n)$ is the half-width of the CI obtained on the basis of n RCs; see Equation (4.3), and $\epsilon(n), 0 < \epsilon(n) < 1$, is the relative statistical error of the CI obtained on the basis of n RCs. As discussed in Section 1.2.1, any sequential experiment using the RC method, with the stopping rule of Equation (4.12), is also stopped at the first checkpoint at which $\epsilon(n) \leq \epsilon_{max}$, where ϵ_{max} is the required limit of the relative statistical error of the simulation results.

From Equation (4.12), we can derive the following formulae for a sequential simulation stopping rule:

$$\frac{\Delta(n)}{\hat{r}(n)} \leq \epsilon_{max} \tag{4.13}$$

or

$$\frac{t_{n-1,1-\alpha/2}}{\epsilon_{max}} \leq \frac{\bar{y}(n)\sqrt{n}}{s(n)}. \tag{4.14}$$

Assuming that we wish to obtain the estimate with a relative statistical error of 5%, at the 95% confidence level, Equation (4.14) reduces, for large samples

[4] A flowchart and pseudocode of the sequential procedure for the RC method are given in Appendix B.3.

$(n \to \infty)$, to

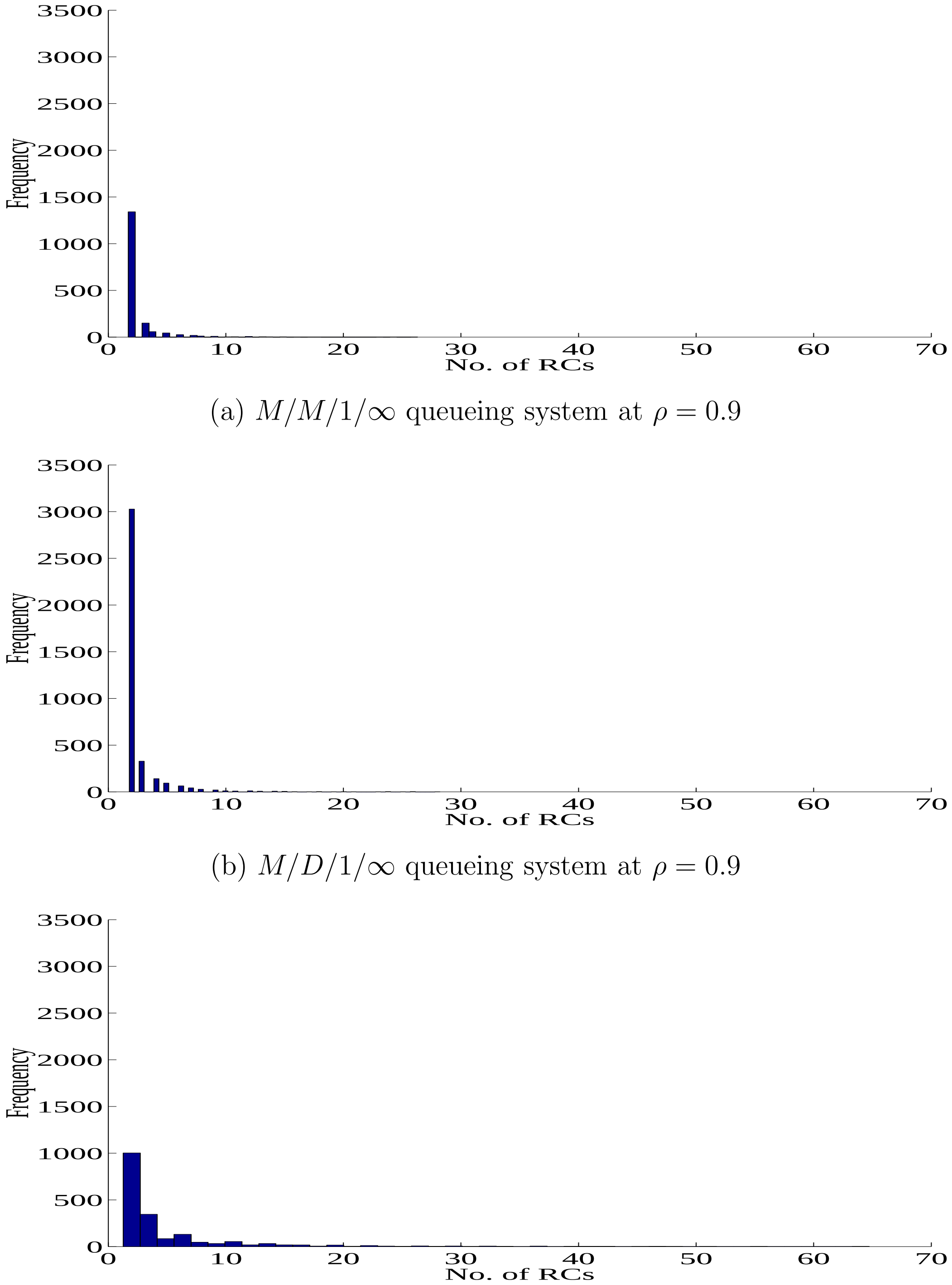
$$\frac{1.96}{0.05} = 39.2 \leq \frac{\bar{y}(n)\sqrt{n}}{s(n)}, \tag{4.15}$$

giving a simpler version of the stopping criterion, which can help us to understand the problem of the sequential method of RCs. This stopping condition can be easily satisfied when an estimated mean (of, for example, waiting times or response times) has a very large value, or its estimated variance $s^2(n)$ has a very small value. Our experiments have shown that after two RCs have been collected (when applying $n_1 = 2$, where n_1 is the run-length measured by the number of RCs, as the location of the first checkpoint), one can sometimes have a very large value of the mean or a very small value of the variance. These situations cause simulation experiments to stop accidentally after too few RCs are collected.

Figures 4.1 (a), (b), and (c) are enlargements of those spikes, which can represent 'extremely short' runs, in Figures 3.2 (c), 3.3 (c), and 3.4 (c) for the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems, respectively. These results clearly show that many sequential simulations are accidentally stopped after as few as two RCs collected, when applying $n_1 = 2$ as the location of the first checkpoint.

These abnormal situations in the sequential method of RCs can happen in practice quite often. We have classified a simulation as 'extremely short' if its recorded RCs is shorter than a threshold, which is 1% of the mean number of collected RCs. Tables 4.1 - 4.3, which are obtained from the same data sets as Figures 3.2 (c), 3.3 (c), and 3.4 (c), show the range of 'extremely short' runs (measured by the number of RCs), the number of 'extremely short' runs, and the probability of dealing with an 'extremely short' run, as well as the mean number of recorded RCs over all 10,000 independent simulation replications and the threshold value for filtering 'extremely short' runs when estimating the mean response time in the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems, respectively. The *n/a* in Table 4.2 means *not applicable*, since none of simulations has less than two RCs.

As reported in the third column of Tables 3.4, 3.7, and 3.10 in Chapter 3,



(a) $M/M/1/\infty$ queueing system at $\rho = 0.9$

(b) $M/D/1/\infty$ queueing system at $\rho = 0.9$

(c) $M/H_2/1/\infty$ queueing system at $\rho = 0.9$

Figure 4.1: Ranges and numbers of 'extremely short' simulation runs observed in the sequential method of RCs when estimating the mean response time at a confidence level of 0.95 with a statistical error $\leq 10\%$

Table 4.1: Statistics from 10,000 replications using the sequential method of RCs with $n_1 = 2$ as the location of the first checkpoint (when estimating the mean response time from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%)

ρ	*Range of extremely short runs*	*Number of extremely short runs*	*Probability of being extremely short runs*	*Mean number of collected RCs per replication*	*Threshold for filtering*
0.1	2 - 4 RCs	567	5.67%	459	4.5
0.2	2 - 5 RCs	601	6.01%	590	5.9
0.3	2 - 5 RCs	593	5.93%	770	7.7
0.4	2 - 5 RCs	598	5.98%	1011	10.1
0.5	2 - 6 RCs	621	6.21%	1371	13.7
0.6	2 - 7 RCs	688	6.88%	1894	18.9
0.7	2 - 7 RCs	797	7.97%	2812	28.1
0.8	2 - 10 RCs	1117	11.17%	4510	45.1
0.9	2 - 28 RCs	1819	18.19%	9296	92.9

short simulation runs collected when applying the sequential method of RCs seriously accelerate the degradation of the quality in terms of coverage, unlike the other analysis methods: NOBM and SA/HW. Most runs among short simulation runs are 'extremely short' in the sequential method of RCs (see Tables 3.4, 3.7, and 3.10 in Chapter 3, and Tables 4.1 - 4.3). This definitely causes very poor coverage, since 'extremely short' simulation runs do not produce valid CIs, which will contain the true value of the parameter, with the specified probability. This also makes the threshold[5] for filtering 'too short' simulation runs much lower than those of the other analysis methods such as NOBM and SA/HW.

All the results so far have used the classical estimator. It is interesting to investigate whether this still occurs with alternative ratio estimators. Therefore,

[5] We assumed the threshold for filtering 'too short' simulation runs calculated from the mean simulation run-length minus one standard deviation of run-lengths, suggested in [134].

Table 4.2: Statistics from 10,000 replications using the sequential method of RCs with $n_1 = 2$ as the location of the first checkpoint (when estimating the mean response time from the $M/D/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%)

ρ	*Range of extremely short runs*	*Number of extremely short runs*	*Probability of being extremely short runs*	*Mean number of collected RCs per replication*	*Threshold for filtering*
0.1	n/a	n/a	n/a	4	0.04
0.2	n/a	n/a	n/a	8	0.08
0.3	n/a	n/a	n/a	15	0.15
0.4	n/a	n/a	n/a	32	0.32
0.5	n/a	n/a	n/a	68	0.68
0.6	n/a	n/a	n/a	152	1.52
0.7	2 - 3 RCs	3226	32.26%	356	3.56
0.8	2 - 8 RCs	3432	34.32%	882	8.82
0.9	2 - 26 RCs	3814	38.14%	2610	26.1

the three selected alternative estimators: the Beale estimator, the jackknife estimator, and the Tin estimator, discussed in Section 4.2, were investigated with including the classical estimator. The distributions of simulation run-lengths, measured by the number of RCs, obtained using all four estimators are depicted in Figure 4.2 using 3,000 independent simulation replications of estimating the mean response time at a traffic intensity $\rho = 0.9$ in the $M/M/1/\infty$ queueing system. We have applied $n_1 = 2$ as the location of the first checkpoint. The distributions obtained are quite similar except for the initial height of the isolated spikes which represent the 'extremely short' simulation runs. The jackknife estimator appears to be the best one, since the height of the spike is the lowest. This may be caused by the theoretical properties of the jackknife estimator. However, even then, 'extremely short' simulation runs still appear.

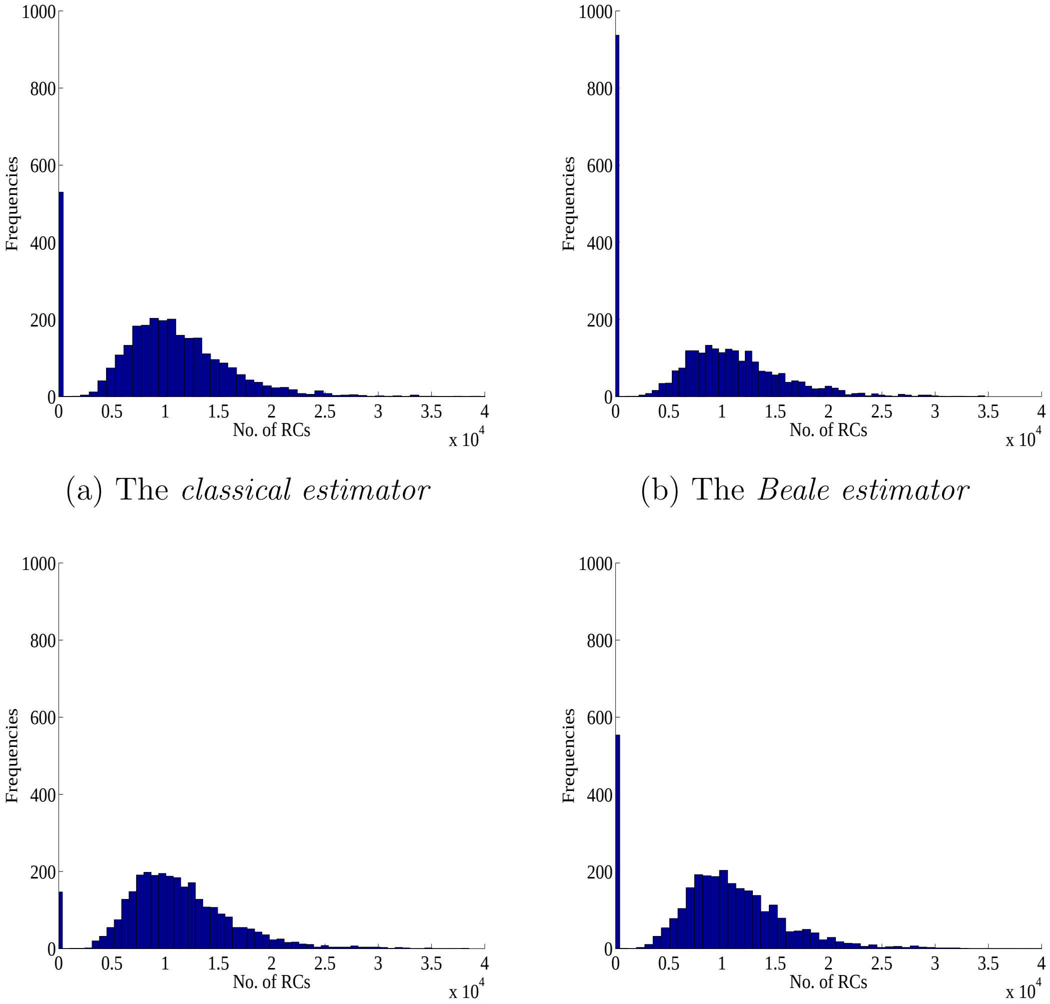
Statistics obtained from the same data sets of Figure 4.2 are also presented in Table 4.4. The threshold for filtering 'too short' runs is the mean number

Table 4.3: Statistics from 10,000 replications using the sequential method of RCs with $n_1 = 2$ as the location of the first checkpoint (when estimating the mean response time from the $M/H_2/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%)

ρ	*Range of extremely short runs*	*Number of extremely short runs*	*Probability of being extremely short runs*	*Mean number of collected RCs per replication*	*Threshold for filtering*
0.1	2 - 4 RCs	443	4.43%	4426	44.2
0.2	2 - 4 RCs	507	5.07%	6941	69.4
0.3	2 - 5 RCs	468	4.68%	9228	92.2
0.4	2 - 4 RCs	503	5.03%	11318	113.1
0.5	2 - 4 RCs	508	5.08%	13461	134.6
0.6	2 - 8 RCs	616	6.16%	15675	156.7
0.7	2 - 18 RCs	806	8.06%	18761	187.6
0.8	2 - 20 RCs	1138	11.38%	23976	239.7
0.9	2 - 64 RCs	1872	18.72%	37334	373.3

of collected RCs minus its standard deviation and the threshold for filtering ‘extremely short’ is defined as runs shorter than 1% of the mean number of collected RCs. The numbers of ‘too short’ runs include the numbers of ‘extremely short’ runs. The results show that the numbers of ‘extremely short’ simulation runs in the sequential method of RCs are very much affected to the mean number of collected RCs per replication and the thresholds used for filtering ‘too short’ runs, since greater numbers of ‘extremely short’ runs cause the mean number of collected RCs per replication and the threshold for ‘too short’ runs to be smaller. This is because all the statistics are obtained from all the executed simulation runs including those that are ‘extremely short’ and ‘too short’.

Of course, filtering simulation runs by discarding those of length shorter than the threshold for ‘too short’ runs does completely remove ‘extremely short’ runs, but it does not remove ‘too short’ runs sufficiently well in most



(a) The *classical estimator*

(b) The *Beale estimator*

(c) The *jackknife estimator*

(d) The *Tin estimator*

Figure 4.2: Distributions of simulation run-lengths, measured by the number of RCs, for the sequential method of RCs using different estimators (when estimating the mean response time at $\rho = 0.9$ from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq$ 10%)

cases. It would seem that the original role of filtering simulation runs with the supposed threshold for 'too short' runs is ineffective in such a case, since many 'extremely short' runs still appear in the sequential method of RCs. As increasing the threshold for filtering 'too short' runs, one can make the role of filtering active. However, it has to be paid the cost of collecting more replica-

Table 4.4: Statistics obtained from 3,000 replications for the sequential method of RCs with different estimators (when estimating the mean response time at $\rho = 0.9$ from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq 10\%$)

Estimators	*Mean number of RCs per replication*	*Threshold for filtering*		*Number of runs filtered*	
		extremely short runs	*too short runs*	*extremely short runs*	*too short runs*
Classical	9,377	93.7	3,231	530	540
Beale	7,848	78.4	1,272	937	938
Jackknife	10,999	109.9	5,771	147	309
Tin	9,279	92.7	3,187	554	564

tions for a sequential coverage analysis or has to use not sufficient number of runs in coverage analysis.

Statistics presented in Table 4.4 also show that the jackknife estimator seems slightly better than the other estimators, since the phenomenon of 'extremely short' runs occurs less often. This makes the threshold for filtering 'too short' runs and the mean number of collected RCs per replication high. Therefore, all observed 'extremely short' runs and many 'too short' runs have been removed in this case. However, no matter which ratio estimator is used, 'extremely short' simulation runs still appear in the sequential method of RCs and affect the final results. Because of this reason and the requirement of extra memory, we do not select the jackknife estimator for the sequential method of RCs. We do select the classical estimator for the sequential method of RCs since it is simple and is recommended for interval estimates in [71].

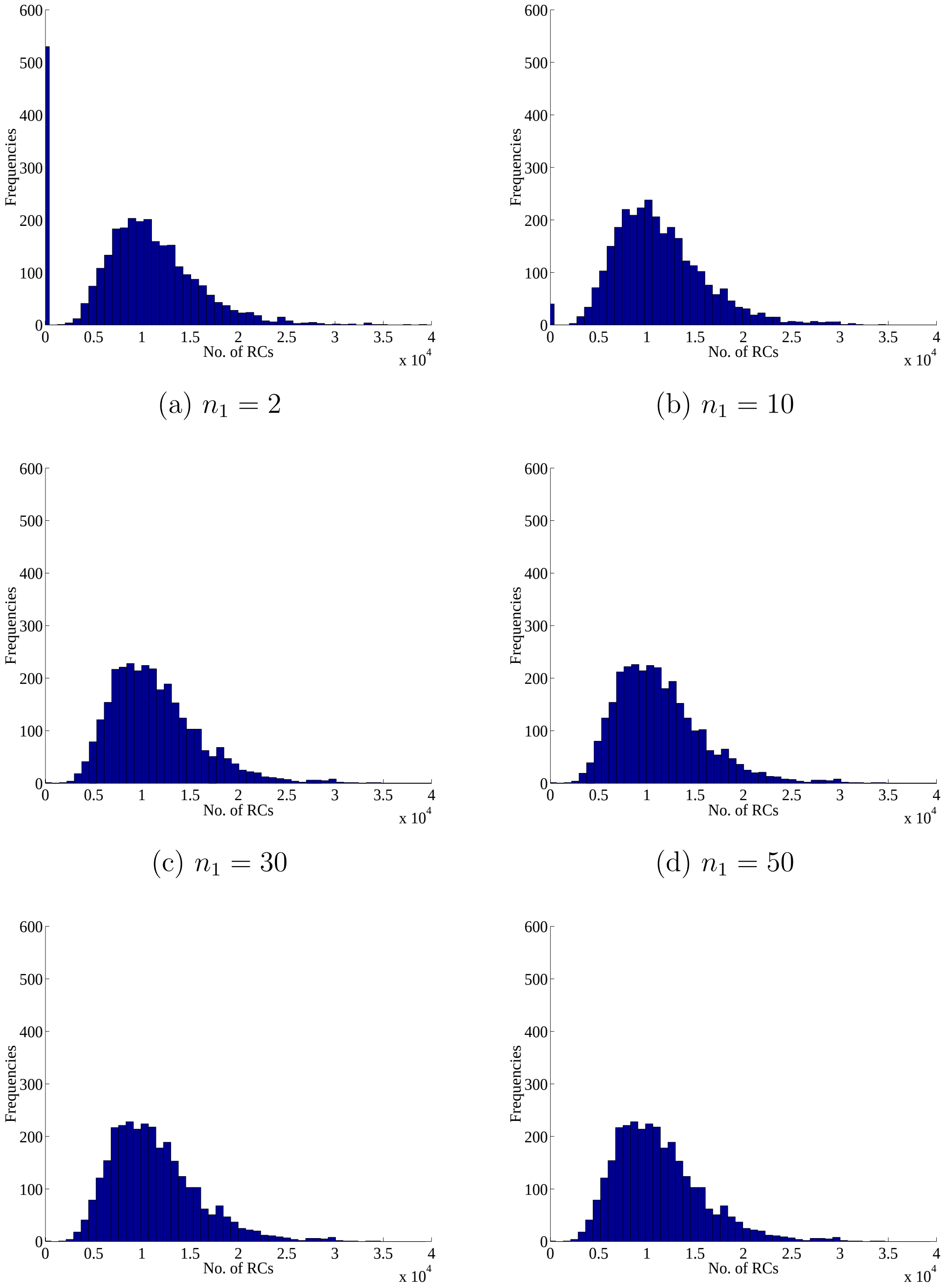
Another solution must be sought to completely remove the 'extremely short' simulation runs. Then, the original role of filtering simulation runs with the threshold for 'too short' runs can be activated to improve the quality of the final results. Therefore, we will investigate the importance of the location of the first checkpoint to find out a solution in Section 4.4.

4.4 Location of the First Checkpoint for the Sequential Method of RCs

The location of the first checkpoint, at which one computes the relative statistical error $\epsilon(n)$ and checks the stopping criterion, can be assumed after collecting at least two RCs (i.e., $n_1 \geq 2$, where n_1 is the run-length measured by the number of RCs) [89], [165]. As shown by the experimental results in Section 4.3, this definitely causes simulation runs are often 'extremely short', since the stopping rules for the sequential method of RCs can be satisfied after collecting only two RCs that are too short. Only, if the location of the first checkpoint is carefully selected, it is possible to avoid collecting 'extremely short' runs.

The distributions of random simulation run-lengths, measured by the number of RCs, for 3,000 independent simulation replications of the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems with a traffic intensity $\rho = 0.9$ are shown in Figures 4.3 - 4.5. A number of different locations (locating it between $n_1 = 2$ to $n_1 = 150$ RCs) of the first checkpoint were assumed. As we would expect, the number of 'extremely short' simulation runs diminishes by delaying the first checkpoint by a larger number of RCs n_1. When a minimum number of 30 RCs or more ($n_1 \geq 30$) for the $M/M/1/\infty$ and $M/D/1/\infty$ queueing systems and 100 RCs or more ($n_1 \geq 100$) for the $M/H_2/1/\infty$ queueing system are assumed, 'extremely short' simulation runs (especially the spike) completely disappear; see Figures 4.3 (c) - (f), 4.4 (c) -(f), and 4.5 (e) and (f). However, 'extremely short' runs can still be seen if a minimum number of ten RCs ($n_1 = 10$), as suggested in [89], are used; see Figures 4.3 (b), 4.4 (b), and 4.5 (b).

The results of coverage analysis with different locations of the first checkpoint ($n_1 = 2$, 10, 30, 50, 100, and 150 RCs) for the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems at a traffic intensity of 0.9, are depicted in Figure 4.6. Each point is obtained from 3,000 independent sequential simulation runs, and the mean response time was estimated at a confidence level of 0.95 with a statistical error less than or equal to 10%. This result shows that



(a) $n_1 = 2$

(b) $n_1 = 10$

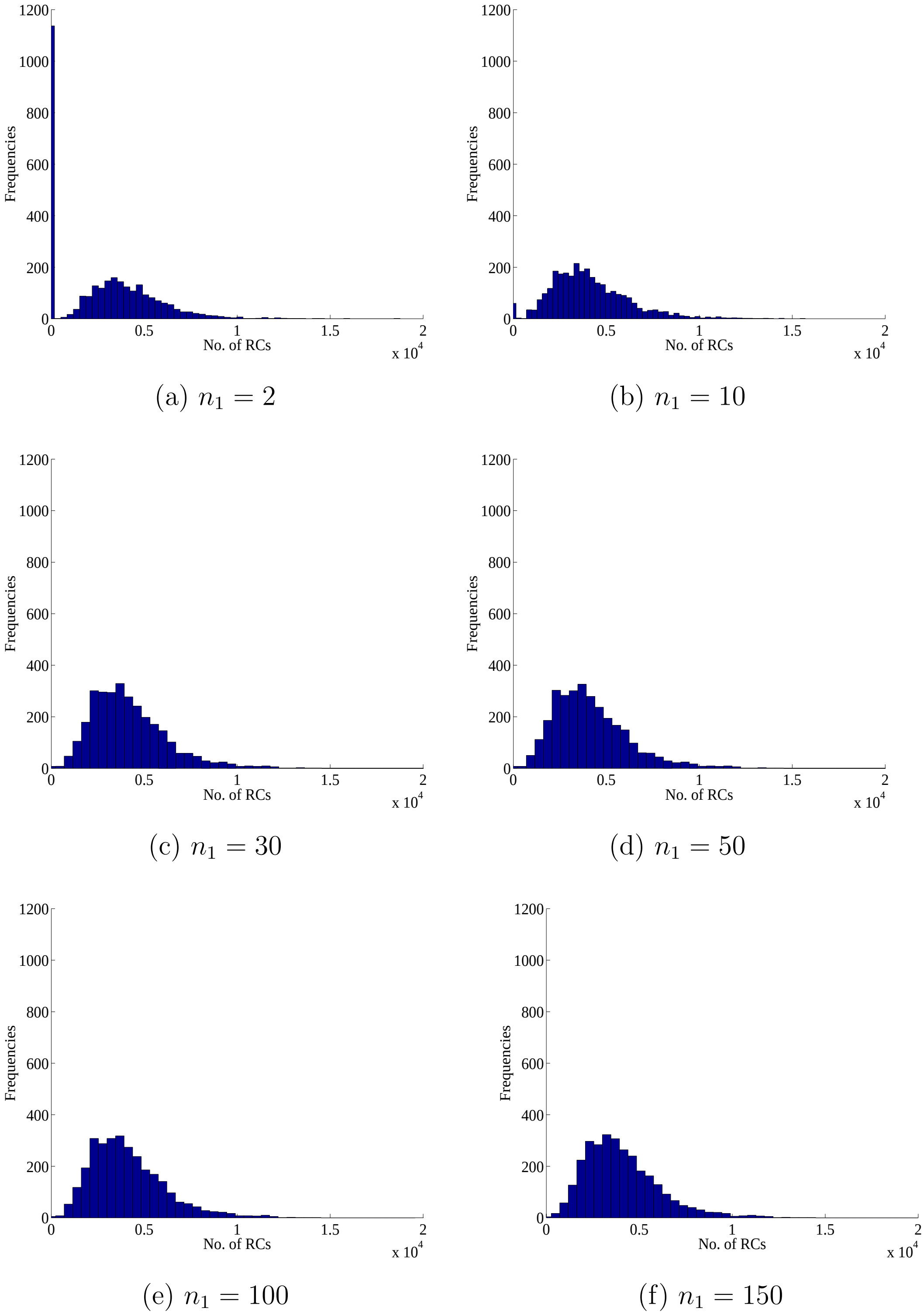
(c) $n_1 = 30$

(d) $n_1 = 50$

(e) $n_1 = 100$

(f) $n_1 = 150$

Figure 4.3: Simulation run-lengths, measured by the number of RCs, with different locations of the first checkpoint ($M/M/1/\infty$, ρ=0.9, n_1 is the location of the first checkpoint)



(a) $n_1 = 2$

(b) $n_1 = 10$

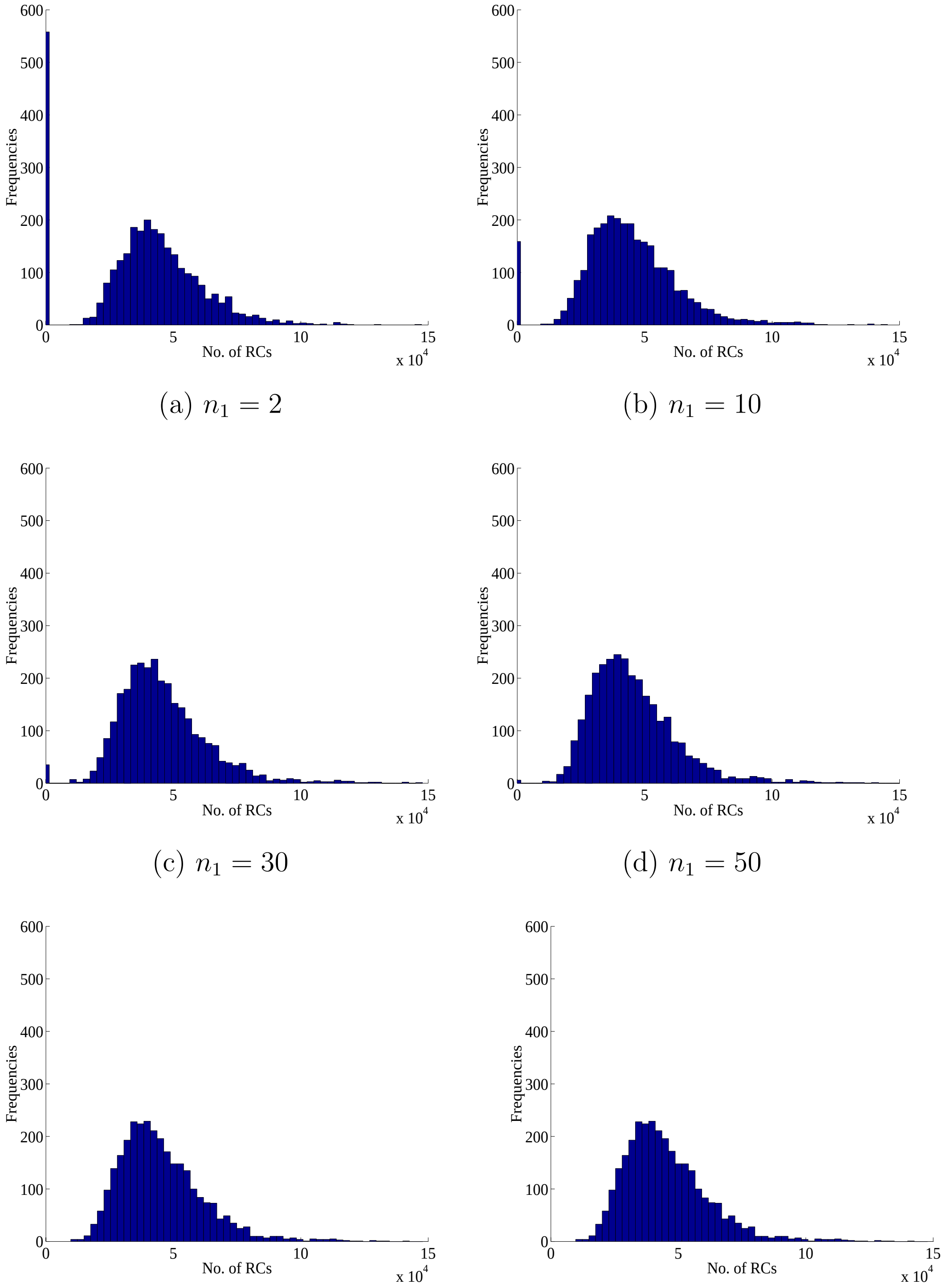
(c) $n_1 = 30$

(d) $n_1 = 50$

(e) $n_1 = 100$

(f) $n_1 = 150$

Figure 4.4: Simulation run-lengths, measured by the number of RCs, with different locations of the first checkpoint ($M/D/1/\infty$, ρ=0.9, n_1 is the location of the first checkpoint)



(a) $n_1 = 2$

(b) $n_1 = 10$

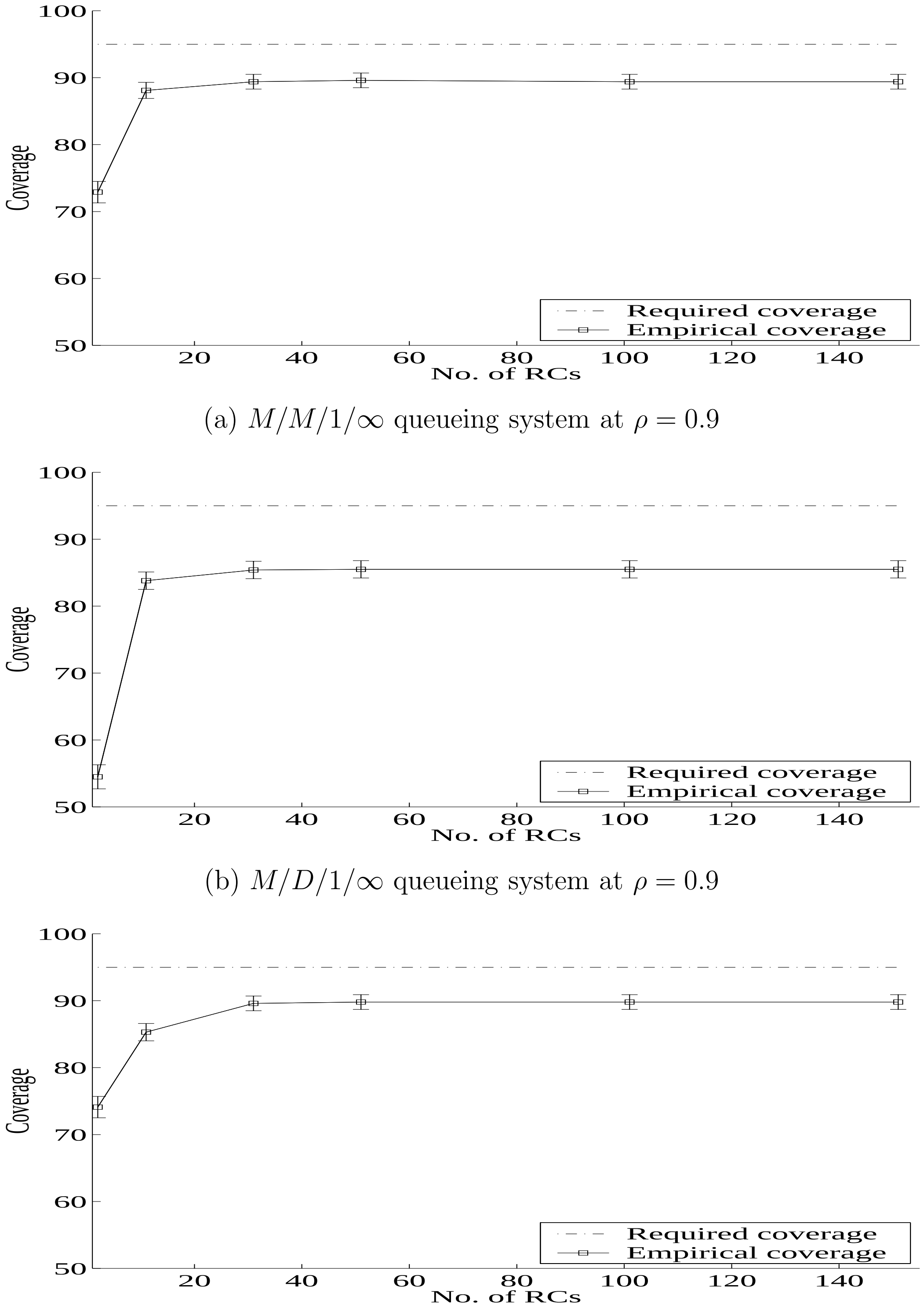
(c) $n_1 = 30$

(d) $n_1 = 50$

(e) $n_1 = 100$

(f) $n_1 = 150$

Figure 4.5: Simulation run-lengths, measured by the number of RCs, with different locations of the first checkpoint ($M/H2/1/\infty$, ρ=0.9, n_1 is the location of the first checkpoint)



(a) $M/M/1/\infty$ queueing system at $\rho = 0.9$

(b) $M/D/1/\infty$ queueing system at $\rho = 0.9$

(c) $M/H_2/1/\infty$ queueing system at $\rho = 0.9$

Figure 4.6: Coverage analysis vs different locations of the first checkpoint n_1 = 2, 10, 30, 50, 100, and 150 RCs (the confidence level = 0.95)

as the location of the first checkpoint is delayed until the predefined minimum number of RCs is collected, the final coverage improves and also converges to a certain level of coverage. This is caused by increasing the simulation run-lengths and decreasing its standard deviations by avoiding the 'extremely short' runs. However, the final coverage is still far from the required level of 0.95.

From these results, we can confirm that more credible final results in the sequential method of RCs can be obtained by choosing a prudent location of the first checkpoint after a suitable number of RCs has been observed. Our results also point to at least 100 RCs[6] or more ($n_1 \geq 100$) as an acceptable location of the first checkpoint in the sequential method of RCs.

As presented in Tables 3.4, 3.7, and 3.10 in Chapter 3, 'too short' simulation runs (including 'extremely short' runs) have poor coverage, especially in the case of the sequential method of RCs, where all are below 21%, compared with the assumed theoretical confidence level of 95%. However, the coverage is significantly improved by the location of the first checkpoint having 100 RCs ($n_1 = 100$), as shown in Tables[7] 4.5 - 4.7 for the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems, respectively. The experimental results of the sequential RCs method are now comparable with NOBM and SA/HW; see Tables 3.2 and 3.3 ($M/M/1/\infty$), Tables 3.5 and 3.6 ($M/D/1/\infty$), and Tables 3.8 and 3.9 ($M/H_2/1/\infty$) in Chapter 3.

Convergences of the coverage of the sequential RCs method when applying the two different locations of the first checkpoint as having two RCs ($n_1 = 2$) and 100 RCs ($n_1 = 100$) are shown in Figure 4.7, for 3,000 independent simulation runs of the $M/M/1/\infty$ queueing system loaded at 0.5. One can see that the final coverage of the sequential method of RCs obtained when applying the location of the first checkpoint as having 100 RCs ($n_1 = 100$) is

[6] For the $M/M/1/\infty$ and $M/D/1/\infty$, 30 RCs or more ($n_1 \geq 30$) as the location of the first checkpoint is an acceptable location, while 100 RCs or more ($n_1 \geq 100$) is an acceptable location for the $M/H_2/1/\infty$. However, we have selected 100 RCs or more ($n_1 \geq 100$), since it is safer to use in practice.

[7] Note that the results for the method of RCs in Tables 4.5 - 4.7 were presented by the number of collected observations to facilitate comparisons with NOBM and SA/HW.

Table 4.5: Sequential method of RCs with the assumption of a minimum number of 100 RCs ($n_1 = 100$) collected before stopping, from 10,000 simulation replications: $M/M/1/\infty$, theoretical confidence level = 0.95

ρ	*Number of too short runs*	*Coverage of too short runs*	*Probability of being too short*	*Threshold for filtering*	*Mean of run-lengths*
0.1	1327	85.9%	13.3%	426	541
0.2	1282	76.6%	12.8%	564	785
0.3	1317	68.0%	13.2%	792	1169
0.4	1226	58.4%	12.3%	1144	1793
0.5	1225	56.2%	12.3%	1795	2922
0.6	1237	52.1%	12.4%	3037	5097
0.7	1286	52.4%	12.9%	6007	10147
0.8	1220	49.4%	12.2%	14787	25675
0.9	1288	49.9%	12.9%	66487	114189

Table 4.6: Sequential method of RCs with the assumption of a minimum number of 100 RCs ($n_1 = 100$) collected before stopping, from 10,000 simulation replications: $M/D/1/\infty$, theoretical confidence level = 0.95

ρ	*Number of too short runs*	*Coverage of too short runs*	*Probability of being too short*	*Threshold for filtering*	*Mean of run-lengths*
0.1	935	34.4%	9.4%	107	112
0.2	135	10.5%	1.1%	113	127
0.3	0	N/A	0.0%	113	156
0.4	0	N/A	0.0%	112	219
0.5	0	N/A	0.0%	129	373
0.6	446	0.0%	4.5%	223	781
0.7	1269	11.7%	12.7%	670	2044
0.8	1254	18.6%	12.5%	2910	7012
0.9	1264	31.6%	12.6%	21423	42596

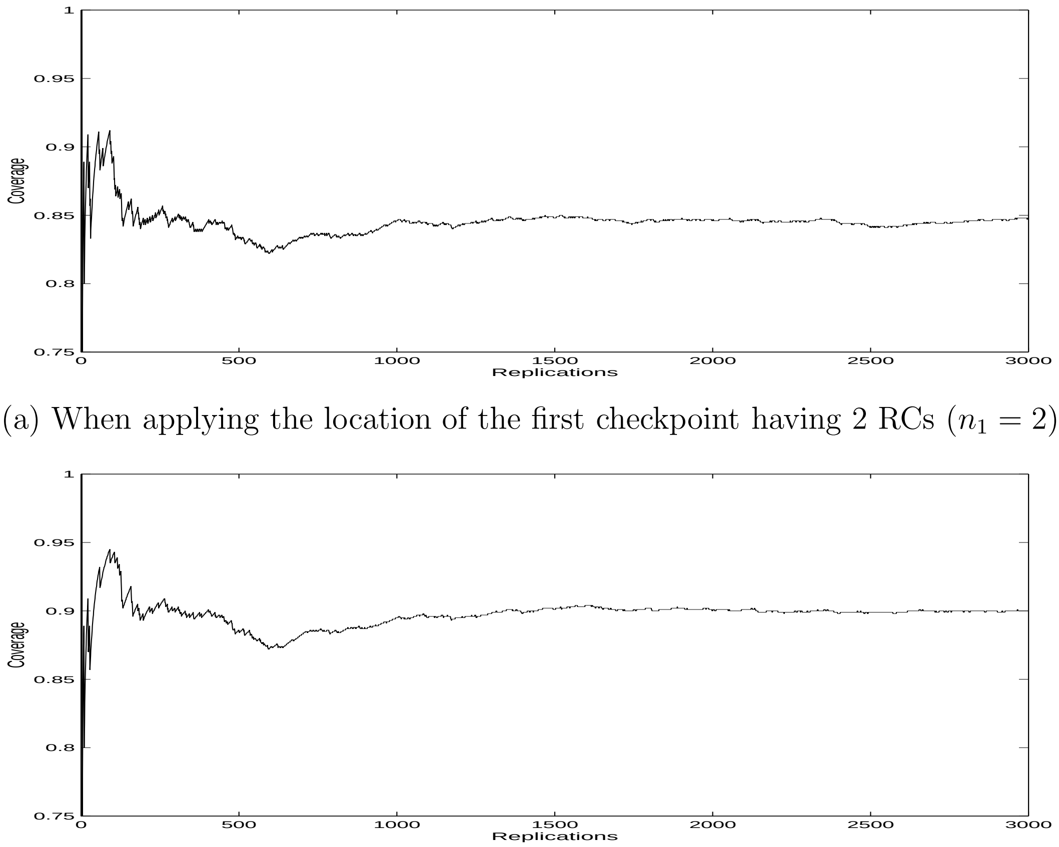
Table 4.7: Sequential method of RCs with the assumption of a minimum number of 100 RCs ($n_1 = 100$) collected before stopping, from 10,000 simulation replications: $M/H_2/1/\infty$, theoretical confidence level = 0.95

ρ	*Number of too short runs*	*Coverage of too short runs*	*Probability of being too short*	*Threshold for filtering*	*Mean of run-lengths*
0.1	1386	72.2%	13.8%	3950	5149
0.2	1342	72.1%	13.4%	6970	9129
0.3	1342	72.4%	13.4%	10550	13840
0.4	1346	75.6%	13.4%	15025	19890
0.5	1360	71.5%	13.6%	21080	28356
0.6	1305	69.1%	13.0%	30257	41887
0.7	1315	68.7%	13.1%	47713	68011
0.8	1335	63.6%	13.3%	90734	135191
0.9	1278	56.9%	12.8%	288628	459121

much better than the one when applying two RCs ($n_1 = 2$).

As discussed in Chapter 2, the CIs of the coverage for any method of simulation output data analysis should contain the confidence level assumed for the final results [156]. In practice, this criterion is seldom met, so it is more appropriate to claim that a method is accepted for practical applications if the CIs of its coverage is sufficiently close to the assumed confidence level. However, Figure 4.7 (a) and (b) show that the final coverage can still be far away from the required level of 0.95, even if the location of the first checkpoint as having 100 RCs ($n_1 = 100$) is applied. As pointed out in [134], this is because an insufficient number of bad final CIs was recorded, as well as the results from 'too short' simulation runs are included in the final results. Applying the location of the first checkpoint as having 100 RCs ($n_1 = 100$) definitely guarantees that the runs of the sequential method of RCs are not 'extremely short'.

Even more accurate results in terms of coverage analysis may be achieved if we additionally adopt the proposed rules (including discarding 'too short'



(a) When applying the location of the first checkpoint having 2 RCs ($n_1 = 2$)

(b) When applying the location of the first checkpoint having 100 RCs ($n_1 = 100$)

Figure 4.7: Convergence of the coverage of the sequential method of RCs (when estimating the mean response time at $\rho = 0.5$ from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq 10\%$)

simulation runs) of experimental coverage analysis for the sequential method of RCs. Therefore, to see an improvement in the final result, we assume that, for representativeness of simulation output data for coverage analysis, a minimum of 200 bad CIs must be recorded before sequential analysis of the coverage can commence and then the results from all simulation runs shorter than a threshold (of mean run-lengths minus one standard deviation of run-lengths) are discarded. Typical convergence of the coverage to its final level for such a scenario is shown in Figure 4.8. As we can see in Figures 4.7 (b) and 4.8, such an approach results in a jump of the coverage from about 0.9 to close to the assumed confidence level of 0.95, as the statistical *noise* introduced by 'too short' simulation runs is removed. From these results, one can see that applying the location of the first checkpoint as having 100 RCs ($n_1 = 100$) and employing the rules of experimental coverage analysis for sequential simulation (discussed in Chapter 2), have great practical value.

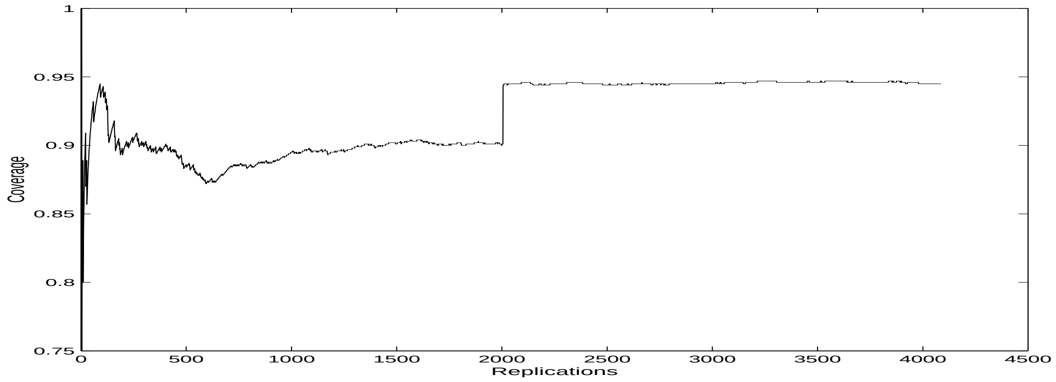


Figure 4.8: Convergence of the coverage of a sequential method of RCs when applying the location of the first checkpoint as having 100 RCs ($n_1 = 100$), collecting at least 200 bad CIs before stopping, and discarding 'too short' runs (when estimating the mean response time at $\rho = 0.5$ from the $M/M/1/\infty$ queueing system at a confidence level = 0.95 with a statistical error $\leq 10\%$)

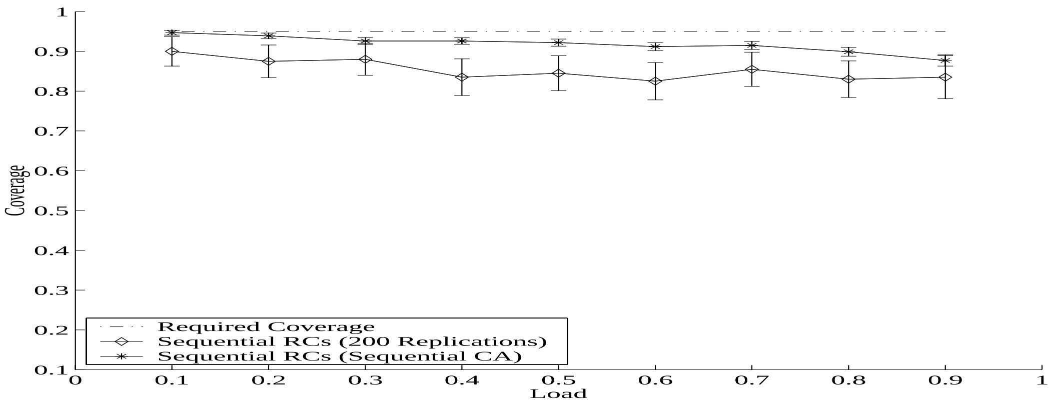
4.5 Coverage Analysis for Sequential Method of RCs

All results of our sequential coverage analysis of the sequential RCs method were obtained assuming the required statistical error of the final result was 5% or less, at a confidence level of 0.95, when applying the rules of experimental coverage analysis for sequential simulation formulated in Chapter 2, and when applying the two different locations of the first checkpoint as having 2 RCs ($n_1 = 2$) and 100 RCs ($n_1 = 100$). The results for the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems obtained from non-sequential coverage analysis of a fixed 200 replications and sequential coverage analysis for the location of the first checkpoint having 2 RCs ($n_1 = 2$) are presented in Figure 4.9. These results show that the final results analysed sequentially are more reliable and credible than the one analysed non-sequentially, since the final result analysed sequentially has higher and more stable coverage, and narrower CIs. However, in heavily loaded systems for the three queueing systems and in lightly loaded systems for the $M/D/1/\infty$ queueing system, the coverage is still not acceptable.

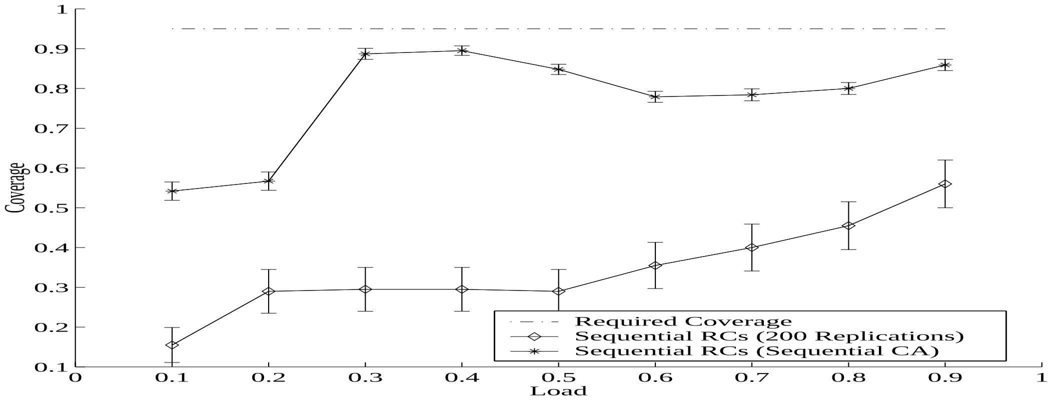
Figure 4.10 shows the results obtained by sequential coverage analysis of the sequential method of RCs when applying the two different locations of the first checkpoint as having 2 RCs ($n_1 = 2$) and 100 RCs ($n_1 = 100$) for the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems, respectively. These results clearly show the remarkable improvement of the quality of the sequential method of RCs when applying the location of the first checkpoint as having 100 RCs ($n_1 = 100$) in the sense of the final coverage and the satisfactorily small statistical errors. The coverage in heavily loaded $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems obtained when applying the location of the first checkpoint as having 100 RCs ($n_1 = 100$) are quite satisfactory, unlike NOBM[8] and SA/HW which are far from the required level; see Figures 2.7 (a) and (b) ($M/M/1/\infty$), Figures 2.8 (a) and (b) ($M/D/1/\infty$), and Figures 2.9 (a) and (b) ($M/H_2/1/\infty$) presented in Chapter 2.

Figure 4.10 (b) also shows that the unusual poor coverage observed in lightly loaded traffic of the $M/D/1/\infty$ queueing system in the RCs method (see also Figure 2.8 (c)) has been significantly improved by assuming the location of the first checkpoint as having 100 RCs ($n_1 = 100$). This clearly means that choosing the proper location of the first checkpoint in the sequential method of RCs is very important.

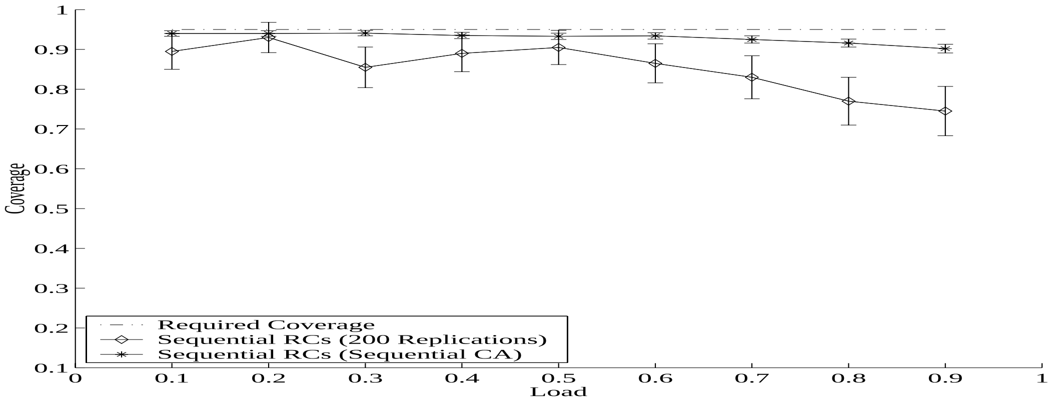
[8] In Akaroa-2 [28], after discarding observations collected during the initial transient period, the locations of the first checkpoint in the sequential NOBM and SA/HW methods are determined by r and $\max(2r, 2n_0)$, where r is the number of batch means of batch size b (the default initial values are $r = 100$, $b = 50$ for NOBM, and $b = 1$ for SA/HW) and n_0 is the length of the initial transient period, respectively; also see [128] for detailed discussion of the first checkpoint for these two methods.



(a) $M/M/1/\infty$ queueing system

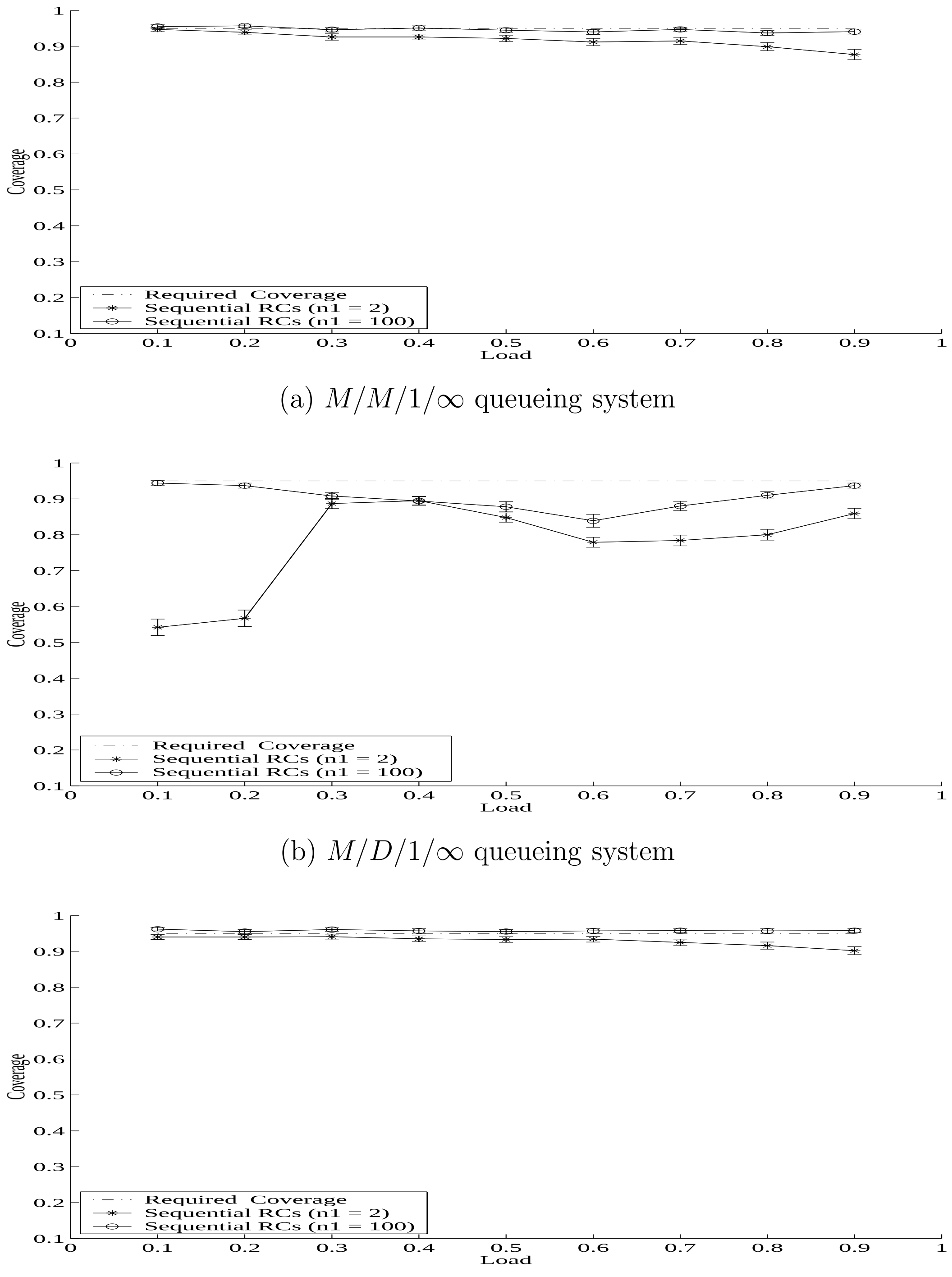


(b) $M/D/1/\infty$ queueing system



(c) $M/H_2/1/\infty$ queueing system

Figure 4.9: Coverage analysis of sequential method of RCs when applying the location of the first checkpoint as having 2 RCs ($n_1 = 2$) (non-sequential coverage analysis of 200 replications and sequential coverage analysis)



(a) $M/M/1/\infty$ queueing system

(b) $M/D/1/\infty$ queueing system

(c) $M/H_2/1/\infty$ queueing system

Figure 4.10: Sequential coverage analysis of sequential method of RCs when applying the two different locations of the first checkpoint as having 2 RCs ($n_1 = 2$) and 100 RCs ($n_1 = 100$)

4.6 Conclusions

Many methods of steady-state simulation output data analysis, such as spectral analysis, batch means, regenerative cycles (RCs), and standardised time series, have been proposed. The method of RCs naturally avoids the problem of the initial transient period. Therefore, for simulation output data analysis this is a very attractive alternative. However, a sequential steady-state simulation with the RC method can lead to inaccurate simulation results if the simulation experiment stops too early, when the sequential stopping criterion is accidentally temporarily satisfied. In particular, 'extremely short' simulation runs observed when applying the sequential method of RCs seriously degrade the quality in terms of coverage, unlike the other analysis methods: NOBM and SA/HW where any 'extremely short' simulation runs have not been observed. 'Extremely short' runs in the sequential method of RCs are caused by an imprudent selection of the location of the first checkpoint. 'Extremely short' runs have as few as two RCs, since the stopping rules have been satisfied at the first checkpoint, when only two RCs are collected. This can happen due to the random nature of the fluctuations in the estimated relative statistical error during the stochastic simulation.

If the first checkpoint for sequential RCs is carefully selected, it is possible to avoid collecting 'extremely short' simulation runs. Lavenberg and Sauer [89] proposed that the simulation should be stopped when a minimum number of RCs is observed (they assumed the arbitrary number of ten) and the required statistical error is obtained. Therefore, we studied the sequential method of RCs with a number of different locations for the first checkpoint. The experimental results clearly show that the number of 'extremely short' simulation runs is diminished by delaying the location of the first checkpoint. As having stopped simulations after a minimum number of 100 RCs or more have been observed, 'extremely short' simulation runs (especially the spike) completely disappear, while 'extremely short' simulation runs can still be seen if a minimum number of ten RCs ($n_1 = 10$) as the location of the first checkpoint, suggested in [89], is used.

Based on our results, we have suggested the best location of the first checkpoint is after a minimum number of 100 RCs or more ($n_1 \geq 100$) have been collected, especially for the sequential RCs method. This enables us to achieve final results with the required statistical error and the required level of coverage, as 'extremely short' and 'too short' runs are eliminated. Adopting $n_1 \geq 100$ as the location of the first checkpoint leads to comparable experimental results with NOBM and SA/HW.

The stopping rules having $n_1 = 100$ as the location of the first checkpoint, and the rules for experimental coverage analysis for sequential simulation proposed in Chapter 2 have also been applied to the analysis of the steady-state means for the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems. These results clearly show that a remarkable improvement in the quality of the sequential method of RCs is obtainable in the sense of the final coverage and the satisfactorily small statistical errors. Our experimental results indicate that the method of RCs in its sequential version is an attractive solution for simulation practitioners if special care is taken to avoid including 'too short' simulation runs, to choose the appropriate location of the first checkpoint to avoid 'extremely short' simulation runs, and to identify the regenerative state.

Chapter 5

AUTOMATION OF SEQUENTIAL STEADY-STATE QUANTILE ESTIMATION

5.1 Quantiles: Importance and Limitations

When simulating a dynamic stochastic system, such as a computer system or telecommunication network, the simulator is frequently more concerned with quantiles which can characterise the extreme performance of the simulated system, than with its average behaviour. Quantiles are particularly useful for planning necessary capacities for various resources, comparing the overall performance of alternative designs, or establishing minimum standards of performance. For example, part of the performance specification for the design of an interactive computer system can be expressed in terms of quantiles of the response time instead of the mean response time, e.g., the 0.9 quantile of the response time should be less than or equal to, say, two seconds. In general, although knowing all the quantiles would be equivalent to knowing the distribution function, one usually looks at only a few quantiles or combinations of quantiles to obtain information about the location, shape, and dispersion of the distribution [109].

We shall first define the concept of quantiles of a distribution of a random variable X. If $F_X(x)$ is a continuous cumulative density function (cdf) and p satisfies $0 < p < 1$, then the equality

$$Pr[X \leq Q_p] = p \tag{5.1}$$

means that Q_p is the p quantile. Intuitively, it means that the random variable X takes values less than or equal to Q_p with probability p. If the cdf is not continuous, Equation (5.1) does not give a quantile for all $p \in (0, 1)$. Then, Q_p is defined as follows:

$$Q_p = \min\{x : Pr[X \leq x] = p\}. \tag{5.2}$$

Notice that Equation (5.2) is also valid for continuous $F_X(x)$ [109].

To estimate quantiles, let $x_1 \leq x_2 \leq \ldots \leq x_n$, $x_i \geq 0$, be the ordered sequence of n observations of a random variable X, collected during the simulation. The usual point estimator of the p quantile, $\hat{Q}_p(n)$, is given by

$$\hat{Q}_p(n) = \begin{cases} x_{np}, & \text{if } np \text{ is an integer} \\ x_{\lfloor np \rfloor + 1}, & \text{if } np \text{ is not an integer} \end{cases} \tag{5.3}$$

where $0 < p < 1$, and $\lfloor np \rfloor$ denotes the integral part of np [109].

For large samples, the estimator $\hat{Q}_p(n)$ performs well, since $(\lfloor np \rfloor + 1)/n \rightarrow p$ as $n \rightarrow \infty$. However, for small samples it may not perform well, particularly when probability p is close to 0 or 1. This may be an important issue if a limited number of observations are available. For such cases, the alternative quantile estimators can be found in [17] and [119]. To efficiently compute quantiles, a few variance reduction techniques, such as antithetic variates ([5]), Latin hypercube sampling ([5]), control variates ([66], [69], [70]), and importance sampling ([48]), have been used to reduce the variance of quantile estimates.

The problem is that when using the estimator of Equation (5.3), especially in the case of correlated sequences of observations, the length of the sample sequence required for achieving an adequately small statistical error of the p quantile $\hat{Q}_p(n)$ can be very large and impossible to predict in advance. Direct application of Equation (5.3) in quantile estimation (QE) requires large amount of computer memory for storing the entire sequence of observations, since this must be sorted whenever a new observation is recorded. The best possible computation time to sort[1] n observations is $O(n \log_2 n)$, and memory proportional to n is required to store sorted values in order to find a given order statistic. This can be a problem. For example, in a steady-state simulation of an $M/M/1/\infty$ queueing system, with traffic intensity $\rho = 0.9$, the estimation of the 0.99 quantile of the waiting times in the queue requires roughly 500,000 observations to achieve an estimate with a relative statistical error of no more than 10% for a 90% CI. For the 0.999 quantile, the required sample size is approximately 2,300,000 to achieve an estimate with a relative statistical error of no more than 10% at 0.9 confidence level [62].

An accurate point estimate of the p quantile $\hat{Q}_p(n)$ could require storing the entire sequence of observations, and its dynamic ordering, as the sequence is expanded. Additional storage would be needed to estimate the variance of the p quantile $\hat{Q}_p(n)$. Clearly, repeated storing and sorting of the entire sequence for QE is impractical in such long runs. Several approaches for estimating quantiles that are linear in computation time and use little memory have been proposed in [72], [74], and [163]. These approaches were originally developed for traditional (non-sequential) procedures. As discussed in Chapter 1, sequential analysis of simulation output data is generally accepted as the only efficient way of achieving an acceptable statistical error of the final results [91]. A sequential QE approach based on a P^2 (*Piecewise-Parabolic*) formula proposed by Jain and Chlamtac [74] has been proposed in [141] and [142].

The most commonly used stopping rule of a sequential stochastic simulation, as discussed in the previous chapters, can be adapted for a sequential QE. Assuming the estimates of an unknown quantile Θ come from a symmetric distribution, the stopping rule is based on the relative half-width of the CIs at

[1] Numerous sorting algorithms are available in the literature, see for example, [67]. Among them, *quicksort* is the best of the sorting methods with regard to the average computing time. With the minor modification of *quicksort* based on the partition algorithm, the p quantile defined in Equation (5.3) can be obtained by a partial sort instead of the full sort of n observations.

a given confidence level, defined as the ratio

$$\epsilon(n) = \frac{\Delta(n)}{\hat{\Theta}(n)}; \qquad 0 < \epsilon(n) < 1, \tag{5.4}$$

where $\hat{\Theta}(n)$ is the point estimate of the unknown quantile Θ from the sequence of n observations and $\Delta(n)$ is the half-width of the CIs for Θ at the (1 - α) confidence level, $0 < \alpha < 1$. In a sequential QE, as with sequential mean estimation, the simulation experiment is also stopped at the first checkpoint at which $\epsilon(n) \leq \epsilon_{max}$, where ϵ_{max} is the required upper limit of the relative statistical error of the final results at the 100(1-α)% confidence level, $0 < \epsilon_{max} < 1$ [100], [103].

Our aim is to find a robust estimator of quantiles which could be used in practical applications of sequential steady-state simulation and could also be implemented in a fully automated simulation package such as Akaroa-2 [28]. In this chapter, we have investigated three sequential QE approaches: *linear* QE (proposed by Iglehart [72] in non-sequential procedures), *batching* QE (proposed by Seila [163], [164] in non-sequential procedures) for the method of regenerative cycles (RCs) in Section 5.2, and *spectral* P^2 QE (proposed by Raatikainen [141], [142] in sequential procedures) for the method of non-RCs in Section 5.3. These do not require storing and sorting of the entire sequence of collected observations. Methods of sequentially detecting the initial transient period for QE are also investigated in Section 5.3. The numerical results of the coverage analysis of these three sequential quantile estimators are presented in Section 5.5. Finally, our findings are summarised in Section 5.6.

5.2 Sequential QE Based on the Method of RCs

Iglehart ([72]) and Seila ([163] and [164]) developed special methods of QE that eliminate the problems of storing and sorting the entire sequences for processes with regeneration points: points at which these processes restart (probabilistically) afresh. An example is the waiting time process in an $M/M/1/\infty$

queueing system which regenerates every time a customer arrives to find the queue empty; see Appendix B.3. Comparisons of their approaches to QE in fixed-sample size stochastic simulations are given in [164].

Here, we consider two sequential quantile estimators for the RC method, based on Iglehart's and Seila's non-sequential proposals, assuming the sequential stopping rule of Equation (5.4) based on the relative statistical error [100], [103]. They are called the *linear* and the *batching* methods of QE. These require $O(n)$ computation time as they do not need sorting, and store only aggregated data for four and two summary statistics, respectively.

5.2.1 Sequential QE Using the *Linear* Approach

The *linear* QE approach was originally developed by Iglehart [72] using the method of RCs with a fixed-sample size. Here, we adapt that approach for sequential QE. First, we specify a grid of h+1 points[2] g_0, g_1, ..., g_h, $g_0 < g_1 < ... < g_h$, so that all observations lie between g_0 and g_h. Next, to find a given quantile estimate, this method estimates the cumulative density function only at the grid points, and uses linear interpolation between them. Simulation continues until the steady-state quantile has been estimated with the required relative statistical error, at the given confidence level. A flowchart of this procedure is given in Figure 5.1. n_1 as the location of the first checkpoint should be selected carefully to produce a sufficient number of observations to ensure that all bins have observations.

Let us consider how the quantile Q_p would be estimated in the course of a simulation experiment by collecting observations in h bins, where an observation is put into bin i if the observation is between grid point g_{i-1} and g_i. Then, having simulated n RCs, we would accumulate the number of observations in each bin. If $w_n(i), i = 1, \cdots, h$, is the total number of observations in bin i during n RCs, then the empirical cumulative distribution function of the

[2] The number of grid points (h+1) and the space between the grid points must be carefully selected to ensure that all observations fall into one of bins and each bin has observations. For example, if the number of grid points (h+1) selected is too large, we can not be sure that each bin contains observations.

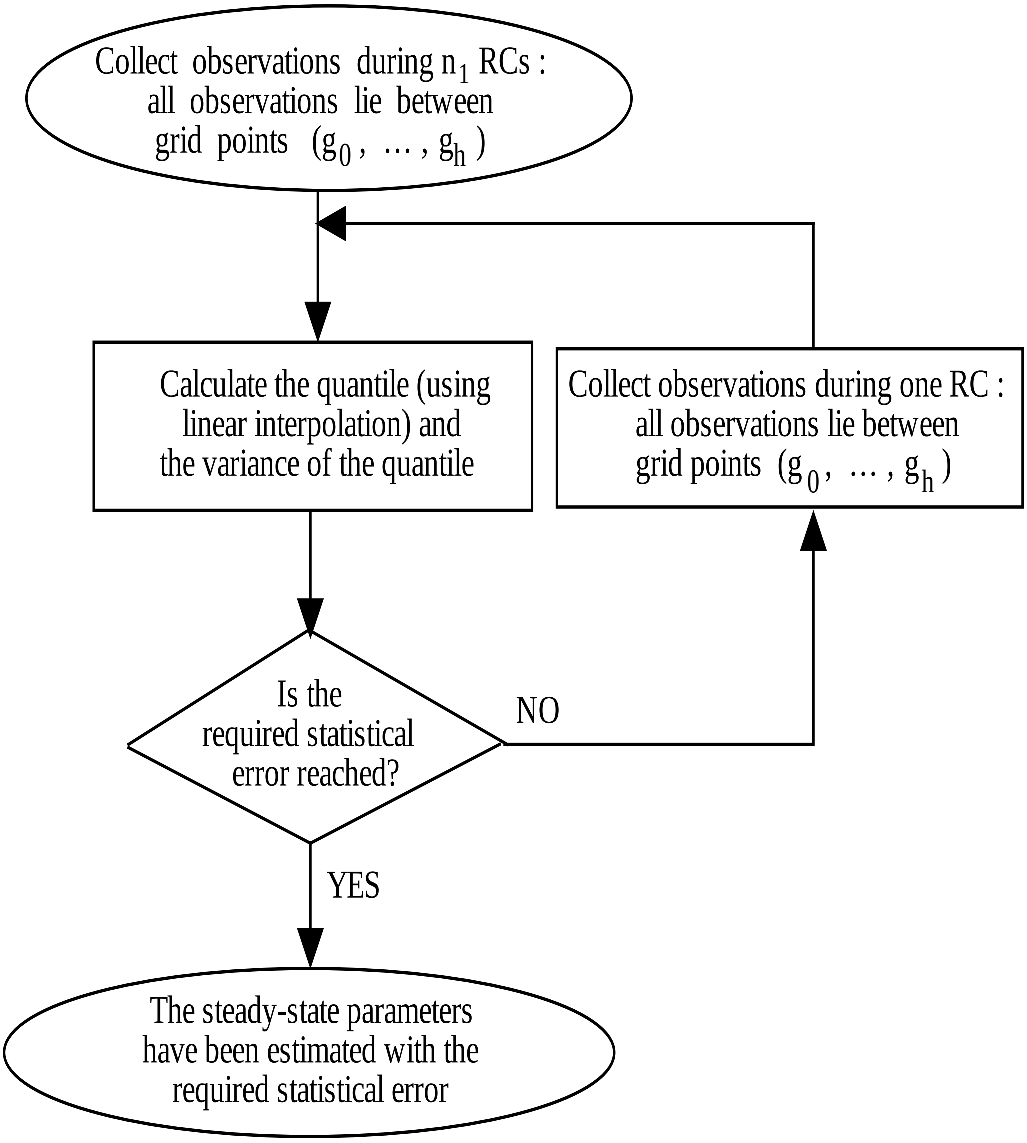

Collect observations during n$_1$ RCs : all observations lie between grid points (g$_0$, ... , g$_h$)
Calculate the quantile (using linear interpolation) and the variance of the quantile
Collect observations during one RC : all observations lie between grid points (g$_0$, ... , g$_h$)
Is the required statistical error reached?
NO
YES
The steady-state parameters have been estimated with the required statistical error

Figure 5.1: Flowchart for the sequential *linear* QE in the method of RCs

random variable X, $F_n(\cdot)$, estimated after n RCs would jump by $w_n(i)/\beta_n$ at grid point g_i, where β_n is the total number of observations collected during n RCs. Then, a new distribution function $\hat{F}_n(g_i)$ at grid point g_i, is estimated by linear interpolation between $F_n(g_i)$ and $F_n(g_{i+1})$. Next, the sample quantile $\hat{Q}_p(n)$ after n RCs would be estimated by taking

$$\hat{Q}_p(n) = \hat{F}_n^{-1}(g_i). \tag{5.5}$$

The variance of this estimator is estimated as

$$\hat{\sigma}^2(\hat{Q}_p(n)) = \hat{\sigma}^2(y_{ij}(n)) - 2F_n(g_i)cov(y_{ij}(n), a_{ij}(n)) + F_n^2(g_i)\hat{\sigma}^2(a_{ij}(n)), \quad (5.6)$$

where $y_{ij}(n)$ and $a_{ij}(n)$ are the sum and the number of observations collected for bin i in the jth RC over n RCs, respectively [72]. Here, $\hat{\sigma}^2(\cdot)$ and $cov(\cdot,\cdot)$ are estimates of the variance and covariance, and $F_n(g_i)$ is the empirical cumulative distribution function of the random variable X after n RCs at grid point g_i.

A $100(1-\alpha)\%$ CI for the quantile Q_p can be obtained by dividing Equation (B.35) in Appendix B.3 with the slope of $F_n(\hat{Q}_p(n))$ [72]. Then, a $100(1-\alpha)\%$ CI for the quantile Q_p is given by

$$\hat{Q}_p(n) \pm \frac{t_{df,1-\alpha/2}\hat{\sigma}(\hat{Q}_p(n))}{\bar{a}F'(\hat{Q}_p(n))n^{\frac{1}{2}}}, \quad (5.7)$$

where $t_{df,1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of the t distribution with $df = n-1$ degrees of freedom, $F'(\hat{Q}_p(n))$ is estimated by $\frac{w_n(\hat{Q}_p(n)+1)}{\beta_n}$, which is the slope of $F_n(\hat{Q}_p(n))$, and $\bar{a} = \frac{1}{n}\sum_{j=1}^{n}(\sum_{i=1}^{h} a_{ij}(n))$; see [72] for more detailed discussion. The pseudocode of the sequential procedures for the *linear* QE approach can be found in Appendix D.1.

5.2.2 Sequential QE Using the *Batching* Approach

The *batching* QE approach was also originally developed for fixed-sample size stochastic simulation only [163] and [164]. First, to adapt it to sequential QE, one needs to group observations from a number of RCs into batches and considers quantile estimates computed for the batches as independent and identically distributed observations. In sequential QE using the *batching* approach, observations collected during a batch have to be sorted. Before applying this method, we must select a batch size b (the number of RCs in a batch), which should sufficiently reduce the cost of computation time in sorting and the memory of storing within the required level of accuracy of the estimate. If the batch size selected is too large, the accuracy of the estimates can be improved and significant data reduction can be achieved. However, the computation time for sorting and the memory for storing observations collected during a batch, can be increased severely. Therefore, the batch size b should be selected to satisfy all the requirements of accuracy, computation time and memory. Seila [163] recommended the batch size of 100 RCs or more to protect against inadequate coverage probabilities. A flowchart of the sequential procedure for the *batching* QE approach is given in Figure 5.2.

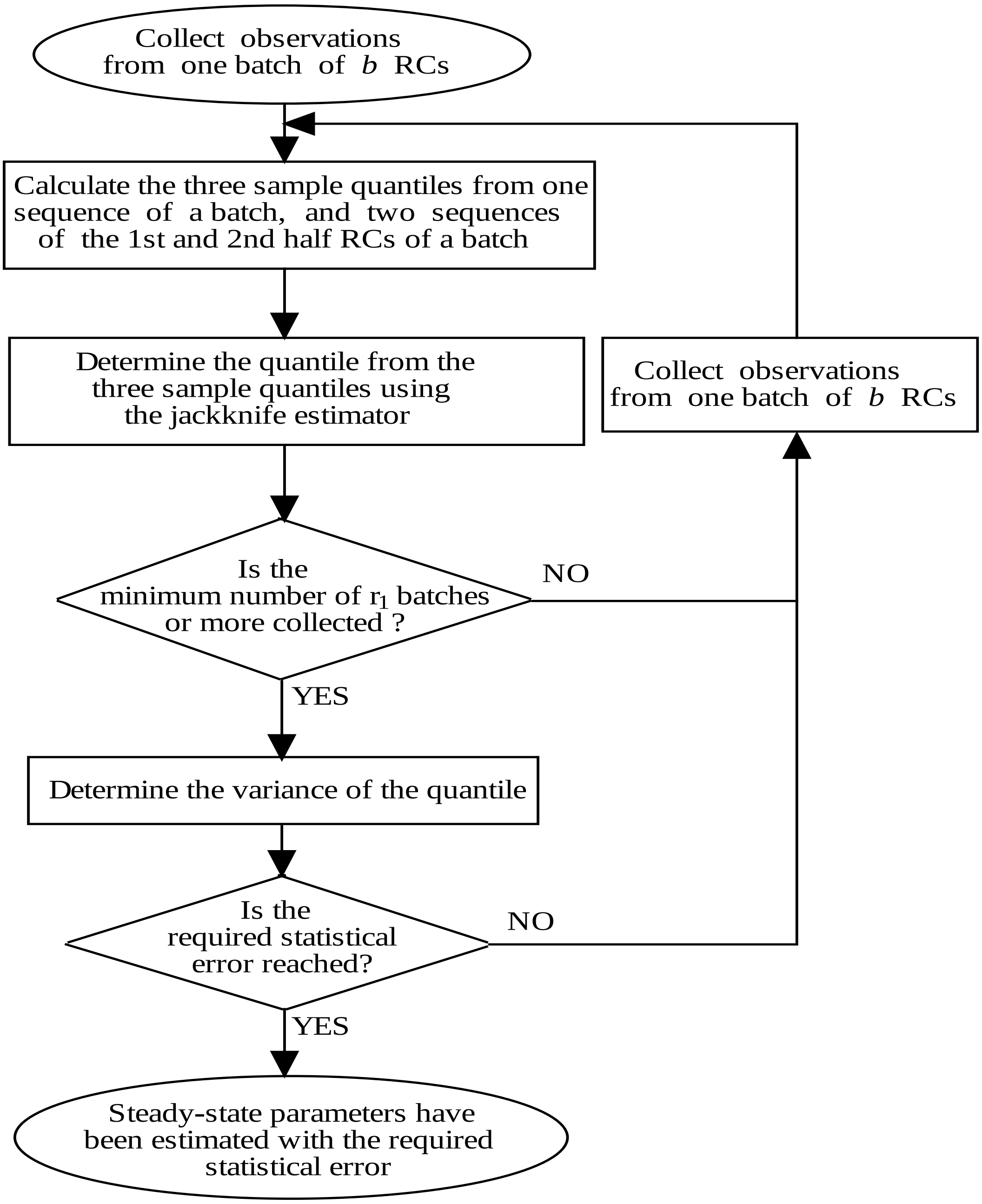

Collect observations from one batch of b RCs
Calculate the three sample quantiles from one sequence of a batch, and two sequences of the 1st and 2nd half RCs of a batch
Determine the quantile from the three sample quantiles using the jackknife estimator
Collect observations from one batch of b RCs
Is the minimum number of r1 batches or more collected ?
NO
YES
Determine the variance of the quantile
Is the required statistical error reached?
NO
YES
Steady-state parameters have been estimated with the required statistical error

Figure 5.2: Flowchart for the sequential *batching* QE in the method of RCs

The *batching* method groups each batch of b RCs, and the three sample quantile estimates are computed from each batch to incorporate a two-fold jackknife procedure in order to reduce bias of the quantile estimators. One sample quantile estimate is computed from all observations collected during a batch, and the other two sample quantile estimates are computed from observations of the first and second half RCs of a batch. Assume that b is even, and let $\hat{Q}_p(b,i)$, $\hat{Q}_p(b/2,i_1)$, and $\hat{Q}_p(b/2,i_2)$ be the estimates of Q_p computed from the b RCs in the ith batch, and the first and second $b/2$ RCs in the ith batch using the ordinary quantile estimator[3], respectively. Then, the jackknifed batch p quantile is

$$J(\hat{Q}_p(b,i)) = 2\hat{Q}_p(b,i) - \frac{1}{2}(\hat{Q}_p(b/2,i_1) + \hat{Q}_p(b/2,i_2)). \tag{5.8}$$

The sequence $\{J(\hat{Q}_p(b,i)), i = 1,2,...,r\}$ over r batches consists of r i.i.d. random variables. Let $J(\bar{Q}_p(b,r))$ and $\hat{\sigma}^2(J(\bar{Q}_p(b,r)))$ denote the mean and variance of such jackknifed quantile estimators, i.e.,

$$J(\bar{Q}_p(b,r)) = \frac{1}{r}\sum_{i=1}^{r} J(\hat{Q}_p(b,i)); \tag{5.9}$$

and

$$\hat{\sigma}^2(J(\bar{Q}_p(b,r))) = \frac{1}{r-1}\sum_{i=1}^{r}(J(\hat{Q}_p(b,i)) - J(\bar{Q}_p(b,r)))^2. \tag{5.10}$$

Then, a $100(1-\alpha)\%$ CI for the quantile Q_p is given by

$$J(\bar{Q}_p(b,r)) \pm \frac{t_{df,1-\alpha/2}\hat{\sigma}(J(\bar{Q}_p(b,r)))}{\sqrt{r}}, \tag{5.11}$$

where $t_{df,1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of the t distribution with $df = r - 1$ degrees of freedom. The pseudocode of the sequential procedures for the *batching* QE approach can be found in Appendix D.2.

[3] The sample quantile is obtained from the order statistic.

5.3 Sequential QE for Non-Regenerative Processes

QE for methods other than those based on RCs have been proposed by Heidelberger and Lewis [62], Jain and Chlamtac [74], and Raatikainen [141], [142]. Heidelberger and Lewis' QE method is based on an aggregation of data sequences by using the maximum transformation, allowing it to work with shorter sequences of (secondary) data. Only storing and sorting of the reduced sequences are needed. They have pointed out the importance of the problem of the initial transient period in the steady-state QE, but it has not been investigated [62].

Jain and Chlamtac's QE method is based on a P^2 (*Piecewise-Parabolic*) formula. The detailed algorithm and its pseudocodes are given in [74]. The P^2 algorithm solves the storage problem by allowing calculations of quantiles dynamically, as the observations are generated. The sequence of observations does not need to be stored. Instead, a few statistical counters are maintained which help to refine the estimate. Therefore, QE using the P^2 algorithm has a very small storage requirement, regardless of the number of observations collected, and a small computing time, because no sorting is required. However, this algorithm has also not considered the problem of the initial transient period of the steady-state estimation.

An *extended* P^2 method based on the P^2 algorithm proposed in [74] has been proposed by Raatikainen [141]. This method simultaneously estimates several quantiles without storing and sorting the observations. A sequential procedure for simultaneous estimation of several quantiles has been proposed in [142]. This sequential version of the *extended* P^2 algorithm for estimating steady-state quantiles uses a spectral method for estimating the variance of the quantile estimates. This procedure has not been equipped with automated detection of the length of the initial transient period. It has been determined by a random number between 1,025 and 2,048 generated using a uniform distribution [142]. In that paper, the numbers of 1,025 and 2,048 were selected for practical reasons only, since the observations were collected in segments and the assumed length of the segment in the sequential version of the *extended* P^2 algorithm was 512.

Recently, another sequential QE approach has been proposed by Chen and Kelton [18]. Chen and Kelton's QE method is based on the order statistics obtained from a pre-specified number of observations to estimate the sample quantile, and the lower and upper bounds of the sample quantile. Storing and sorting of the pre-specified number of observations are needed. However, once the lower and upper bounds of the sample quantile are obtained, other observations do not need to be stored. A newly generated observation will then only be stored in the available memory when the observation is between the lower and upper bounds inclusively. Therefore, this method reduces both the memory requirement and the computation time. They have adopted the stopping rule of absolute statistical error for a sequential QE. However, the problem of the initial transient period of the steady-state estimation has not been considered.

Therefore, firstly we will discuss detection methods of the initial transient period for the steady-state estimation of quantiles, and investigate two detection methods that should perform well (see [16], [57], and [81]) in a fully automated simulation package [104]. Then, sequential QE using the *spectral* P^2 approach ([100], [103]), not based on RCs, will be discussed, together with the best detection method of the initial transient period.

5.3.1 Detection Methods of the Initial Transient Period for QE

In steady-state simulations of non-regenerative processes, the performance after the system has reached a stable state is of interest. In such cases, results of the initial transient period of the simulation should not be included in the final results. If the initial transient period is not discarded properly, then the inclusion of these observations in an estimate can lead to a serious bias in that estimate, known as initialisation bias [104].

One way of dealing with initialisation bias is to run the simulation experiments for a sufficiently long period to make any influence of the initial transient period negligible. While such an approach to a stochastic steady-state simulation can sometimes lead to acceptable results, one may still finish with statistically inaccurate results since it is difficult to ensure that the length of run chosen is long enough.

A more appropriate method is to collect observations only after the system has reached steady-state. This may completely eliminate the initialisation bias. However, a problem with this approach is that one needs to recognise that steady-state has been achieved. Of course, if the output is truncated too early, then significant bias might still be present. If it is truncated too late, then many good observations are lost.

A number of ways to estimate the length of the initial transient period of steady-state simulations for estimators of mean values have been proposed in [16], [57], and [162]. Basic problems related to the existence of initial transient periods can be found, for example, in [128], [150], and [180]. The length of the initial transient period has traditionally been determined using different heuristic rules. A survey of heuristic rules can be found in [128]. More precise measures of the length of the initial transient could be obtained by using various statistical tests invented to test the stationarity of data sequences. These tests operate in a hypothesis testing framework, formally testing the null hypothesis that *there is no initialisation bias in the output mean* against the alternate hypothesis that initialisation bias exists in the output.

Numerous statistical tests have been proposed by Goldsman, Schruben and Swain [57], Schruben [160], and Yücesan [182]. Comparative studies can be found in [16], [57], and [81]. Their studies revealed that two statistical tests proposed by Schruben et al. [162], and Goldsman, Schruben and Swain [57] can determine the length of the initial transient period quite well [104].

All heuristic rules and statistical tests for detecting the initial transient period have been developed for the case where the steady-state mean of the system is estimated, but none have been developed for estimating steady-state quantiles. Most papers discussing QE have not considered the problem of the initial transient period; see [18], [62], and [74], with the exception of one written by Raatikainen [142]. However, Raatikainen [142] has discarded an initial transient period of random length (determined by the uniform distribution U(1025, 2048)) to reduce the initialisation bias. This is definitely better than no consideration of the initialisation bias, but it is not the best idea.

So far no theory has been developed on the rate of convergence of quantiles to their steady-state values. However, from studies of the convergence of quantiles to theoretical quantiles (see for example, M. Fisz [36], pp. 377 - 379), it appears that sample quantiles should converge stochastically to their limit values in much the same way as sample means. A quantile is also closely related to the probability of a level; see Equations (5.1) and (5.2). In the case of a symmetric continuous distribution (e.g., normal distribution), for example, a 50th percentile (e.g., 0.5 quantile) of the parameter of interest is equal to the sample mean of the parameter of interest [109]. Therefore, this suggests that one could apply statistical tests developed for the mean to detect the length of the initial transient period of QE. We adopt two statistical tests proposed by Schruben [162] and Goldsman, Schruben and Swain [57] to discover the suitability of applying them in the case of QE. These are briefly summarised below.

Schruben's Test

Stationarity tests, based on a standardised time series estimator of mean value, known as the maximum estimator, were first proposed by Schruben in [160]. These were improved by Schruben et al. in [162] using one of the standardised time series estimators called the area estimator; see Appendix C for detailed discussion of the two estimators. The latter test, based on the area estimator ([162]), will be called Schruben's test in this dissertation.

Schruben's test is based on the asymptotic convergence of partial sums of deviations to a limiting stochastic process called the 'Brownian bridge' $\{B_t; 0 \leq t \leq 1\}$, i.e., a model of Brownian motion on the unit interval conditioned to start and return to zero. It is used to test the hypothesis that a sufficient number of initial transient observations has been discarded. Rejection or acceptance of the hypothesis that the given sub-sequence of observations is stationary, or equivalently, that the initial transient period is not included in collected observations, depends on the probability characterising the value calculated from the considered sequence. This test is quite simple numerically, and can be applied to a wide class of simulated processes.

A practical problem faced when implementing one of these tests is that they require *a priori* knowledge of the steady-state variance σ^2 of the simulated process, which is not normally available when the test is applied, because the system is still in its initial transient period. These tests solve this problem by estimating the steady-state variance over the latter portion of the collected data [160], [162]. This is done on the assumption that this latter portion of data is more representative of the steady-state behaviour of the system, thus giving a better estimate of the steady-state variance. The effectiveness of the test is strongly dependent on how accurately the variance estimator is estimated.

Goldsman, Schruben and Swain's Test

Goldsman, Schruben and Swain [57] discussed a few statistical tests, based on the different variance estimators of the batch means, the area estimator, the maximum estimator, and also combinations of these estimators, for detecting the initial transient period. Cash et al. [16] have studied these statistical tests and recommended that based on the maximum estimator. Therefore, this will be considered as a candidate method for detecting the initial transient period for QE in a fully automated simulation package. This test will be named the GSS test after its authors.

The GSS test is a natural generalisation of the test proposed by Schruben in [160]. In this test, observations $x_1, x_2, \cdots, x_n$ are divided into r batches of length b (assume $n = rb$). The variance estimator based on the first r' batches is compared to the corresponding estimator from the remaining $r - r'$ batches. The null hypothesis of *no initialisation bias in the output mean exits* is rejected if $F > F_{1-\alpha,3r',3(r-r')}$. Here, $F = V_{r'}/V_{r-r'}$, where $V_{r'}$ and $V_{r-r'}$ are the variance estimators from the first r' batches and the last $r - r'$ batches, respectively, and $F_{1-\alpha,3r',3(r-r')}$ is the $1-\alpha$ quantile of an F distribution with $3r'$ and $3(r-r')$ degrees of freedom.

For the GSS test, the compromise choice of the number of batches $r = 8$ with a sufficiently large batch size b and the number of batches in the first portion $r' = 6$ were recommended in [16]. If these estimators are deemed to be significantly different, then an initial transient mean is assumed to be still present.

5.3.2 Comparisons of Two Statistical Tests for QE

Schruben's test as the initial transient period detection method, based on the area estimator for estimating the variance of the sample mean $\overline{X}(n)$ (see Appendix C), has been proposed in [162] and implemented in the simulation package Akaroa-2 [28]. When estimating the steady-state mean, the initial transient period is automatically and sequentially detected.

There is a simple check to determine whether this method is suitable for sequential QE in non-regenerative processes. Just after the initial transient period, the value of (say) the 0.9 quantile of the waiting time in the queue for an $M/M/1/\infty$ queueing system with a traffic intensity of $\rho = 0.8$ should be close to its theoretical steady-state 0.9 quantile, which can be calculated by

$$\max\left(0, \frac{E[w]}{\rho}\ln[10\rho]\right), \tag{5.12}$$

where $E[w]$ is the theoretical mean waiting time in the queue [73], [104].

Equation (5.12) has been derived in the following way. The cumulative distribution function of the waiting times can be shown to be

$$F(w) = 1 - \rho e^{-w\mu(1-\rho)}. \tag{5.13}$$

This is a (defective) exponential distribution. From the distribution, we can find out its quantiles. For example, the p quantile of the waiting time (w_p) can be computed as follows:

$$1 - \rho e^{-w_p\mu(1-\rho)} = p \tag{5.14}$$

or

$$w_p = \frac{1}{\mu(1-\rho)} \ln\left(\frac{\rho}{1-p}\right). \tag{5.15}$$

This formula applies only if p is greater than $(1-\rho)$. All lower quantiles are zero. This can be stated in one equation as follows:

$$w_p = \max\left(0, \frac{E[w]}{\rho} \ln\left[\frac{\rho}{1-p}\right]\right) \tag{5.16}$$

where the mean waiting time in the queue $E[w]$ is $\rho/(\mu(1-\rho))$. From Equation (5.16), we can have the simplified version for the 0.9 quantile (Equation (5.12)) [73].

Figure 5.3 shows the theoretical convergence of the waiting time of the n-th customer (in the $M/M/1/\infty$ queueing system at a traffic intensity of 0.8) to the theoretical steady-state waiting time. The theoretical steady-state is calculated using Equation (5.16). The theoretical waiting time of the n-th customer is calculated using the algorithm proposed by McNickle [115]. As the number of customers n is increased, the waiting times converge to the

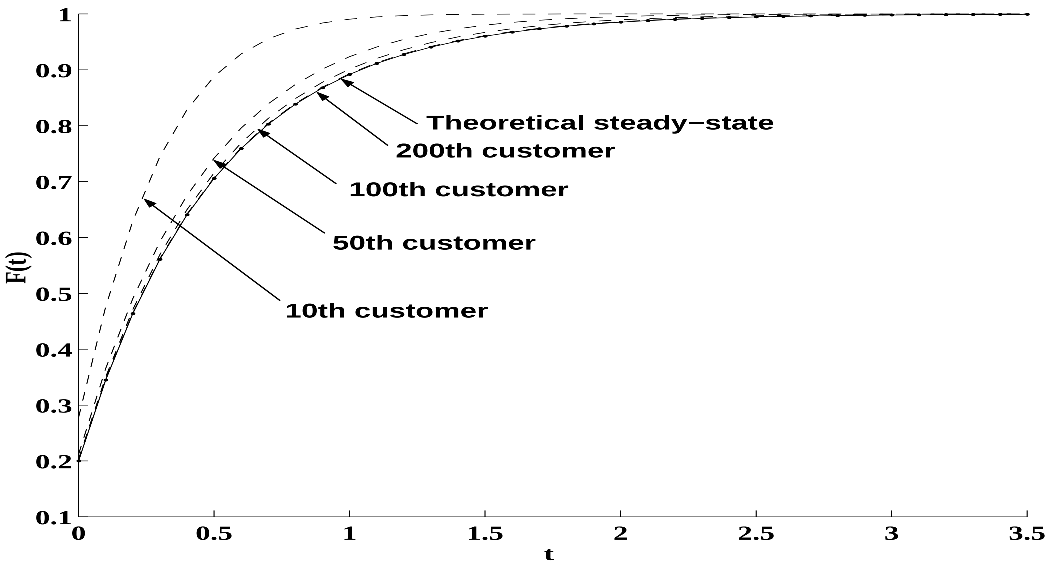


Figure 5.3: Theoretical convergence of the cumulative distribution function of waiting times of the n-th customer in the $M/M/1/\infty$ queue to the theoretical steady-state (at a traffic intensity of $\rho = 0.8$)

theoretical steady-state. For example, the mean waiting time of the 300th customer at a traffic intensity of $\rho = 0.8$ is within -0.0739% of the steady-state mean, while the 0.9 quantile of the 300th customer at a traffic intensity of $\rho = 0.8$ is within -0.032% of the steady-state 0.9 quantile. Therefore, if the empirical quantiles have similar convergence to the theoretical convergence as shown in Figure 5.3, the initial transient detectors in the estimator of the mean may also work moderately well for estimating steady-state quantiles. The influence of even a mis-estimated initial transient period in QE can be limited, since QE involves quite long runs of 8,681 ± 221 observations[4].

If the theoretical and empirical values are dissimilar, some of the results may be biased since initialisation bias still exists even after deleting the observations collected in the initial transient period. The results of the validation, obtained using the Schruben test, are depicted in Figure 5.4 together with the

[4] This range of run-length is obtained from 6,000 independent replications when estimating the 0.9 quantile of response times in the $M/M/1/\infty$ queueing system at a traffic intensity of $\rho = 0.9$ using the *spectral* P^2 approach.

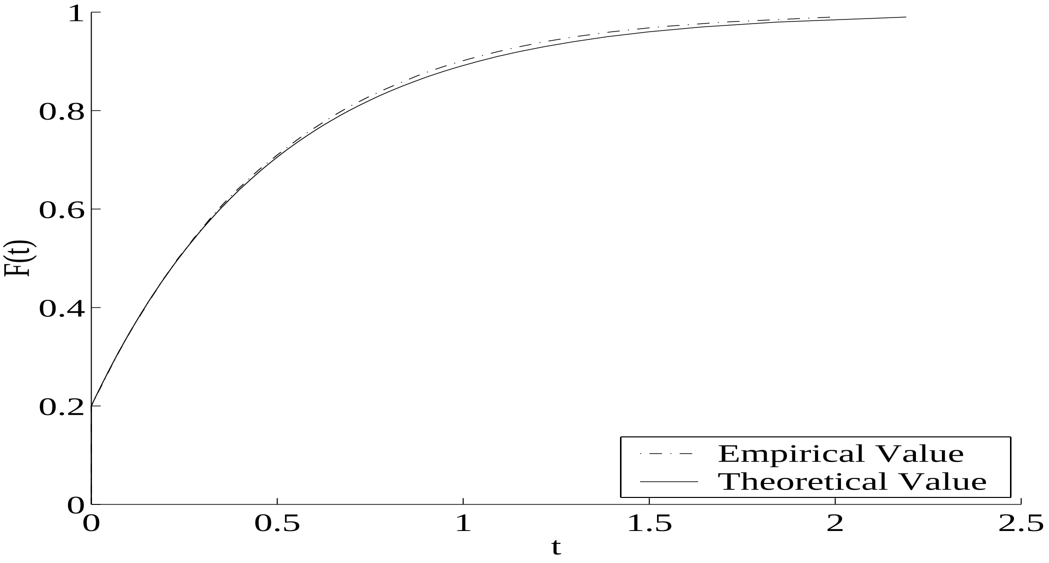


Figure 5.4: Comparisons of theoretical and empirical values of cumulative distribution function of waiting times in the queue when using the Schruben test for detecting an initial transient period ($M/M/1/\infty$ queueing system at a traffic intensity of $\rho = 0.8$)

theoretical values calculated using the Equation (5.16). To obtain the empirical quantiles, we executed 100,000 independent simulation runs and measured the waiting time of the first recorded customer (after the initial transient period) from each simulation run for 100,000 independent replications. The 0.9 quantile of the waiting time in the queue of the first recorded customer (after the initial transient period) was calculated and compared with the theoretical steady-state quantiles calculated using Equation (5.12). This simple experiment shows that the 0.9 quantile of the waiting time in the queue is quite close to its theoretical steady-state 0.9 quantile. The 0.9 quantile is within -4.541% of the steady-state 0.9 quantile. This experiment is simple, but at least gives some justification for using the detection method of the initial transient period, originally developed for the estimator of means, when estimating the steady-state quantiles in the methods based on non-RCs [104].

Secondly, to find a better statistical test for detecting an initial transient period for steady-state quantiles, we have also investigated the performance of the GSS test based on the maximum estimator of the standardised time series which was reported as giving the best performance in Cash et al. [16] and Goldsman et al. [57]. The empirical results of the GSS test are depicted in Figure 5.5 together with the theoretical values calculated using Equation (5.16). To obtain the empirical quantiles, we also executed 100,000 independent simulation runs and measured the waiting time of the first recorded customer (after the initial transient period) from each simulation run for 100,000 independent replications. In this case, the 0.9 quantile is within +6.65% of the steady-state 0.9 quantile. The convergence of the waiting time to the theoretical steady-state should follow the result shown in Figure 5.3. However, the empirical results of the GSS test do not follow. When also comparing the results of the Schruben and GSS tests (shown in Figure 5.4 and Figure 5.5, respectively), the GSS test clearly shows considerably worse performance than Schruben's test, since the empirical values are much larger than the theoretical values over the (almost) entire range of quantiles [104].

Another comparison of the Schruben and GSS tests is presented in Table

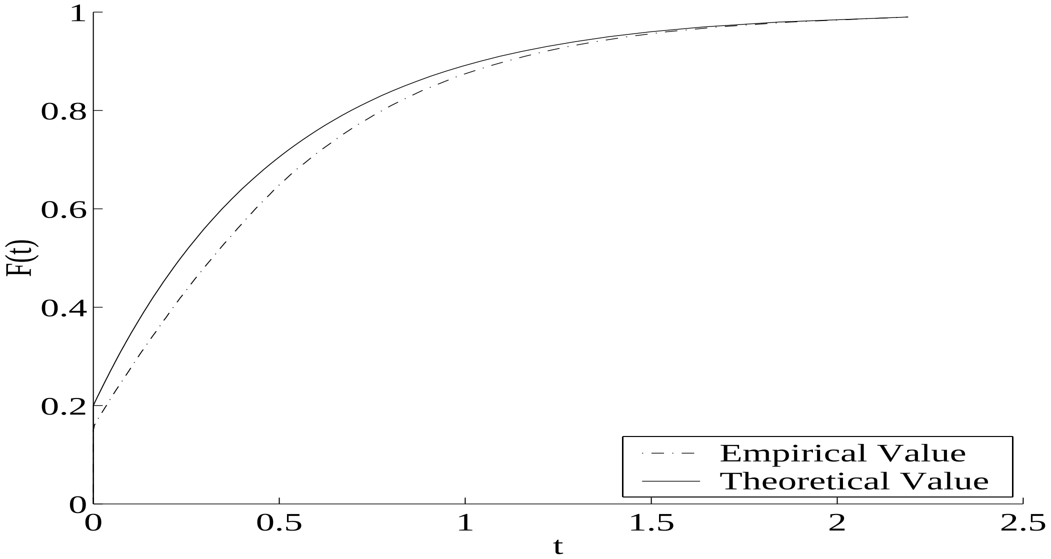


Figure 5.5: Comparisons of theoretical and empirical values of cumulative distribution function of waiting times in the queue when using the GSS test for detecting an initial transient period ($M/M/1/\infty$ queueing system at a traffic intensity of $\rho = 0.8$)

5.1[5]. The statistical data is obtained from the same data as in Figures 5.4 and 5.5. The mean value of the waiting time in the queue of the first recorded customer (after the initial transient period) and the mean number of transient observations obtained from 100,000 independent replications are presented in Table 5.1 (I). As we can see, the mean of the waiting times obtained from the Schruben test is closer to the theoretical steady-state mean waiting time than the mean of the waiting times obtained from the GSS test. The two tests have detected longer initial transient periods than the period required theoretically, but the detected initial transient periods with the Schruben test are a little longer than the GSS test. This suggests that Schruben's test is better than the GSS test in detecting the length of the initial transient period when estimating

[5] The theoretical steady-state mean waiting time in the queue for the $M/M/1/\infty$ queueing system equals $\rho/(\mu(1-\rho))$ [73]. The procedure of calculating the theoretical number of transient observations required when estimating the waiting time can be found in [115]. If we make the assumption that we are in steady-state when the mean waiting time is very close (within 0.05%) to the steady-state value, then (using the algorithm proposed in [115]) we can achieve steady-state within 0.05% after 326 customers.

Table 5.1: Statistics obtained from the Schruben and GSS tests for detecting the initial transient period in the steady-state estimation of means ($M/M/1/\infty$ queueing system at the 0.95 confidence level with the required relative statistical error of 10% or less)

(I)

	Mean of Waiting Times	**Mean of Transient Observations**
Theory	0.4	326
Schruben Test	0.384 ± 0.0024	455.74 ± 2.83
GSS Test	0.457 ± 0.0028	372.56 ± 2.31

(II)

Quantile	**Theory**	**Schruben Test**	**GSS Test**
0.5	0.2350	0.2336 (-0.596%)	0.3197 (+36.04%)
0.9	1.04999	0.9925 (-5.475%)	1.1089 (+5.610%)
0.95	1.3903	1.2999 (-6.502%)	1.4368 (+3.344%)
0.99	2.1910	1.9921 (-9.078%)	2.1900 (-0.045%)

(III)

ρ	**GSS Test (372)**		**Schruben Test (455)**	
	Mean	**0.9 Quantile**	**Mean**	**0.9 Quantile**
0.6	$-6.3e-11$ %	$+2.024e-4$ %	$-2.5e-13$ %	$+2.024e-4$ %
0.7	$-3.38e-6$ %	$+1.233e-4$ %	$-1.81e-7$ %	$+1.251e-4$ %
0.8	$-2.48e-2$ %	$-1.36e-2$ %	$-7.14e-3$ %	$-7.55e-3$ %
0.9	-5.3168 %	-5.9082 %	-3.6169 %	-4.0526 %

mean values.

Quantiles at 0.5, 0.9, 0.95, and 0.99, obtained for the Schruben and GSS tests from the results of Figures 5.4 and 5.5, are presented in Table 5.1 (II) with the theoretical values. The errors of the mean and 0.9 quantile of the waiting time to the theoretical steady-state obtained at the traffic intensities of 0.6 to 0.9 using the algorithm proposed in [115], when the lengths of the initial transient period are assumed to be 372 (observed for GSS test) and 455 (observed for Schruben Test), are also presented in Table 5.1 (III). As expected, the errors of estimates of steady-state mean and 0.9 quantile of the waiting times have decreased if longer length of the initial transient period is assumed. The differences between the estimates of 0.9 quantile and the theoretical value are larger than differences in means, but they converge in a similar way as the autocorrelations increase. This agrees well with the experimental evidence shown in Figure 5.4. Therefore, the initial transient detectors developed for the case where the steady-state mean of the system is estimated can be applied when estimating steady-state quantiles, since no better detection method of the initial transient period for QE is available. The Schruben test again appears to be better than the GSS test, since it detects the reasonably longer end point of the initial transient period, which produces smaller errors to the theoretical steady-state.

5.3.3 Sequential QE Using the *Spectral* P^2 Approach

Here, we consider a fully automated sequential procedure for QE, which we will call *spectral* P^2. The length of the initial transient stage is automatically determined using the Schruben test ([162]), which is quite a reasonable method for detecting the initial transient period in a sequential QE as discussed in Section 5.3.2. Then steady-state QE begins, and stops when the relative statistical error reaches the required level. The *spectral* P^2 QE approach is based on the P^2 algorithm proposed by Jain & Chlamtac [74] for estimating quantiles, and on the formula given in Raatikainen [140], [142], modified from the SA/HW method (originally proposed for estimating the variance of the mean in [63]), for estimating the variance of the quantile estimate . When the output sequence of n observations is stationary and satisfies the ϕ-mixing condition[6], the quantile estimate $\hat{Q}_p(n)$, based on the order statistic, has a normal limiting

[6] The ϕ-mixing property informally states that if the process runs for a sufficiently long time, observations in the distant past are approximately independent of those in the present [93].

distribution. These asymptotic properties of the order statistics in estimating the variance of the quantile estimate are well summarised in [142].

A flowchart of the sequential procedure of the *spectral* P^2 QE in the method based on non-RCs is given in Figure 5.6. The location of the first checkpoint and next checkpoints have been determined by following the procedure discussed in [27] and [128]. The location of the first checkpoint is determined by

$$w_1 = \max[200, 2 * n_0], \tag{5.17}$$

where n_0 is the number of observations collected from the initial transient period. The next checkpoints after the first checkpoint are determined by using the *linear* spacing method. In *linear* checkpoint spacing, the distance between successive checkpoints is determined as a multiple of the length of the initial transient period by

$$w_i = 2 * n_0 * SpacingFactor \tag{5.18}$$

[27]. We have assumed the SpacingFactor 1.5. The pseudocode can be found in Appendix D.3.

Having collected n observations, the p quantile $\hat{Q}_p(n)$ estimated by the P^2 algorithm is actually approximated from the inverse of the empirical cumulative distribution function by a piecewise-parabolic formula. The P^2 algorithm consists of maintaining the five markers. Each marker has a height, which is equal to the estimation of a specific quantile, an actual position, and a desired position. The parabolic formula assumes that the curve passing through any three adjacent markers is a parabola of the form $q_i = an_i^2 + bn_i + c$, where q_i is the height and n_i is the actual position of the i-th marker. That is, one can have the following three equations:

$$\begin{aligned} q_{i-1} &= an_{i-1}^2 + bn_{i-1} + c, \\ q_i &= an_i^2 + bn_i + c, \quad and \end{aligned} \tag{5.19}$$

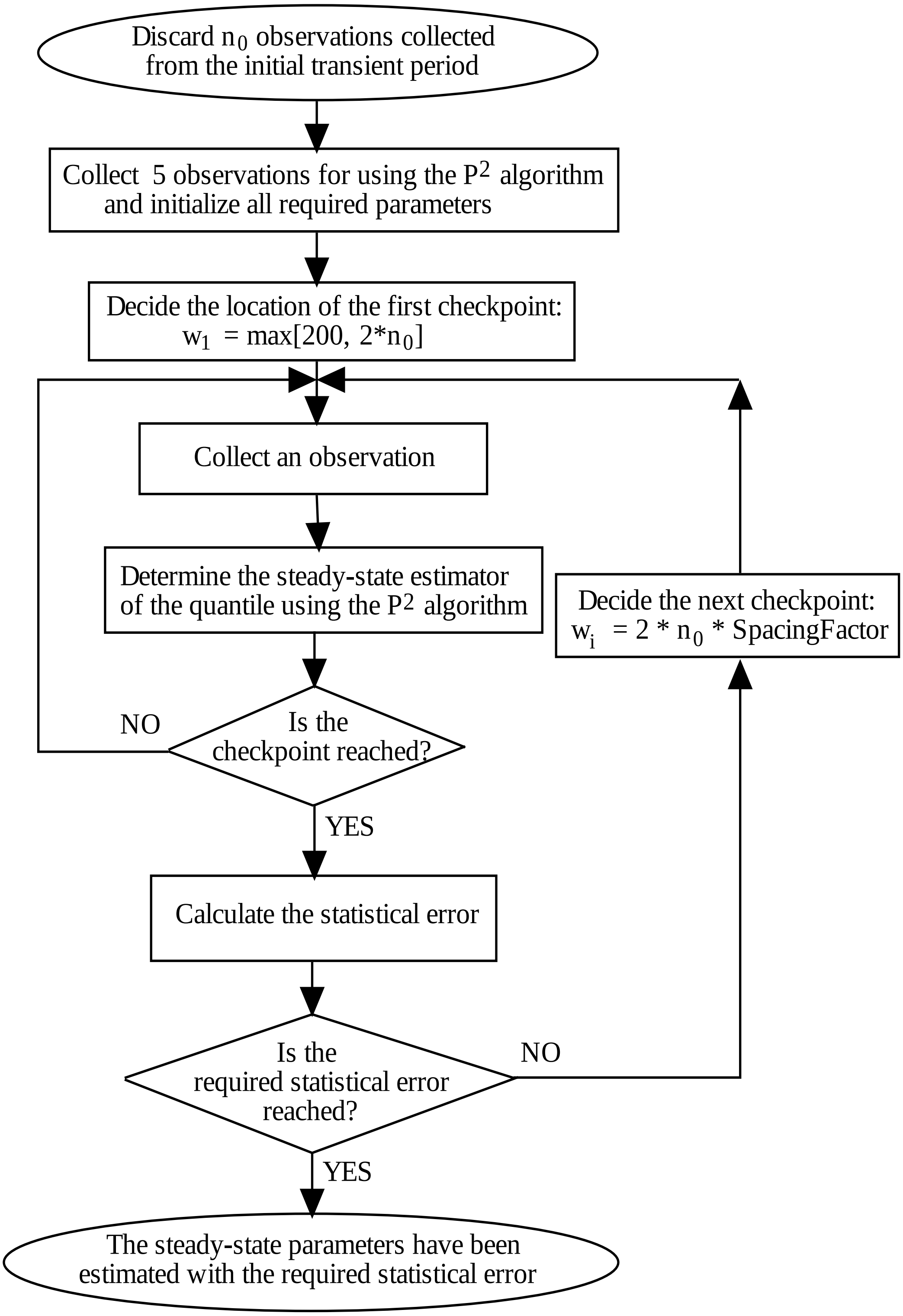

Discard n0 observations collected from the initial transient period
Collect 5 observations for using the P2 algorithm and initialize all required parameters
Decide the location of the first checkpoint: w1 = max[200, 2*n0]
Collect an observation
Determine the steady-state estimator of the quantile using the P2 algorithm
Is the checkpoint reached?
NO
YES
Calculate the statistical error
Is the required statistical error reached?
NO
YES
Decide the next checkpoint: wi = 2 * n0 * SpacingFactor
The steady-state parameters have been estimated with the required statistical error

Figure 5.6: Flowchart for the method of sequential *spectral* P^2 QE in the method of non-RCs

$$q_{i+1} = an_{i+1}^2 + bn_{i+1} + c.$$

The coefficients a, b, and c are determined by solving the above three equations. Then, a marker height at $n_i^{'} = n_i + d$, where $d = \pm 1$, is adjusted using either the parabolic formula of

$$q_i^{'} = q_i + \frac{d}{n_{i+1} - n_{i-1}} \left\{ (n_i - n_{i-1} + d)\frac{q_{i+1} - q_i}{n_{i+1} - n_i} + (n_{i+1} - n_i - d)\frac{q_i - q_{i-1}}{n_i - n_{i-1}} \right\}, \tag{5.20}$$

or the linear formula of

$$q_i^{'} = q_i + d\frac{q_{i+d} - q_i}{n_{i+d} - n_i}. \tag{5.21}$$

The parabolic formula is usually used, but the linear formula is sometimes used to keep the marker heights in an increasing order. Finally, the height of the third marker q_3 is the estimate of the p quantile $\hat{Q}_p(n)$: see [74] for detailed discussion.

The variance of the quantile estimate $\hat{Q}_p(n)$ is estimated by using the formula given in [142]. As the number of observations becomes large, the variance of $\hat{Q}_p(n)$ can be approximated by

$$\sigma^2(\hat{Q}_p(n)) = \frac{S(0; \hat{Q}_p(n))}{n\hat{f}^2(\hat{Q}_p(n))}, \tag{5.22}$$

where $S(0; \hat{Q}_p(n))$ is the spectral density at frequency 0, estimated using the SA/HW method proposed by Heidelberger and Welch [63] (see also Appendix B.2), and $\hat{f}(\hat{Q}_p(n))$ is the empirical density function, approximated by $\hat{f}(\hat{Q}_p(n)) = (b + 2a\hat{F}(\hat{Q}_p(n)))^{-1}$ since $\hat{Q}_p(n)$ is an approximation of the inverse of the empirical cumulative distribution function, $\hat{F}^{-1}(n) = an^2 + bn + c$ [142].

A $100(1-\alpha)\%$ CI for the quantile Q_p is given by

$$\hat{Q}_p(n) \pm \frac{t_{df,1-\alpha/2}\hat{\sigma}(\hat{Q}_p(n))}{\sqrt{n}}, \tag{5.23}$$

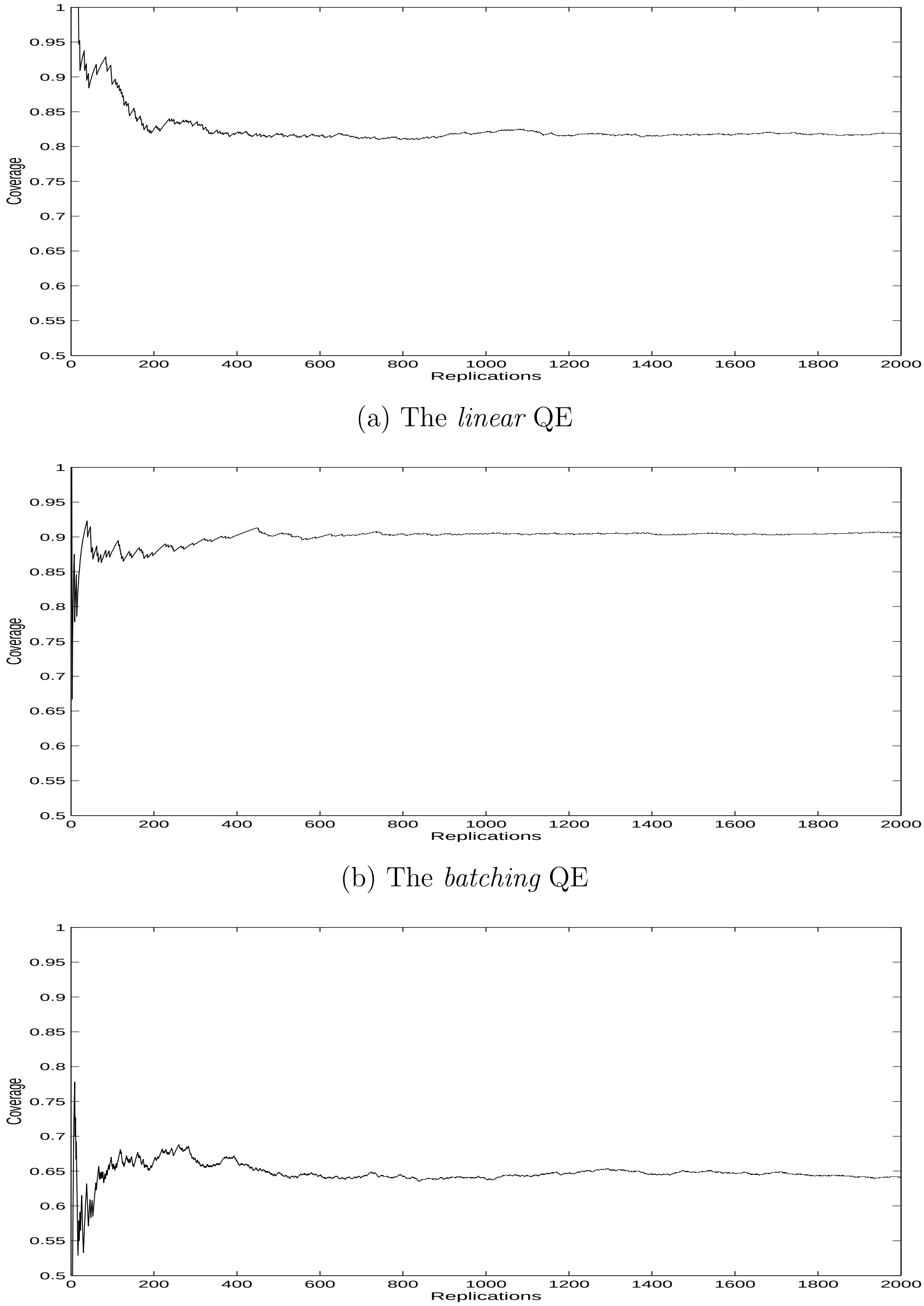
where $t_{df,1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of the t distribution with degrees of freedom $df = 7$.

5.4 Experimental Coverage Analysis for Sequential QE

The robustness of each estimator can be measured experimentally by the coverage of its CIs as discussed and justified in Chapter 2. We have checked whether the rules of experimental coverage analysis for sequential simulation formulated in Chapter 2 give similar effects for QE. Firstly, the convergences of coverage for the three different approaches of QE: *linear* QE, *batching* QE, and *spectral* P^2 QE, are depicted in Figure 5.7. (We have obtained results as assuming $n_1 = 1,000$ RCs as the location of the first checkpoint and 21 grid points spaced 0.2 units apart for the *linear* QE approach, and $r_1 = 10$ batches as the location of the first checkpoint and a batch size of 100 RCs for the *batching* QE approach. The reason for choosing these values will be discussed in the next section.) These results also show a high initial instability of coverage as in the mean value estimation; see Figure 2.5 in Chapter 2.

The convergences of coverage for the three different approaches of QE when applying the rules of experimental coverage analysis for sequential simulation in Chapter 2, are depicted in Figure 5.8. These results, obtained after filtering unusually short simulation runs, which are not reliable, show a clear improvement of the final coverage as in the mean value estimation; see Figure 2.6 in Chapter 2. The coverage analysis was stopped when a relative statistical error of at least 5% at the 0.95 confidence level was reached and the recommended (in [134]) 200 bad CIs (i.e., CIs that do not cover the theoretical value) were collected.

Having compared the results presented in Figures 5.7 and 5.8, one can also see the importance of applying appropriate method of coverage analysis in the case of estimating quantiles. Note that sequential analysis of coverage not only leads to more accurate results, but it also allows for full automation of the tedious comparative studies of properties of different estimators. Therefore, we will apply these rules of experimental coverage analysis (from Chapter 2) with the same conditions to the three different approaches of QE: *linear* QE, *batching* QE, and *spectral* P^2 QE, in Section 5.5.

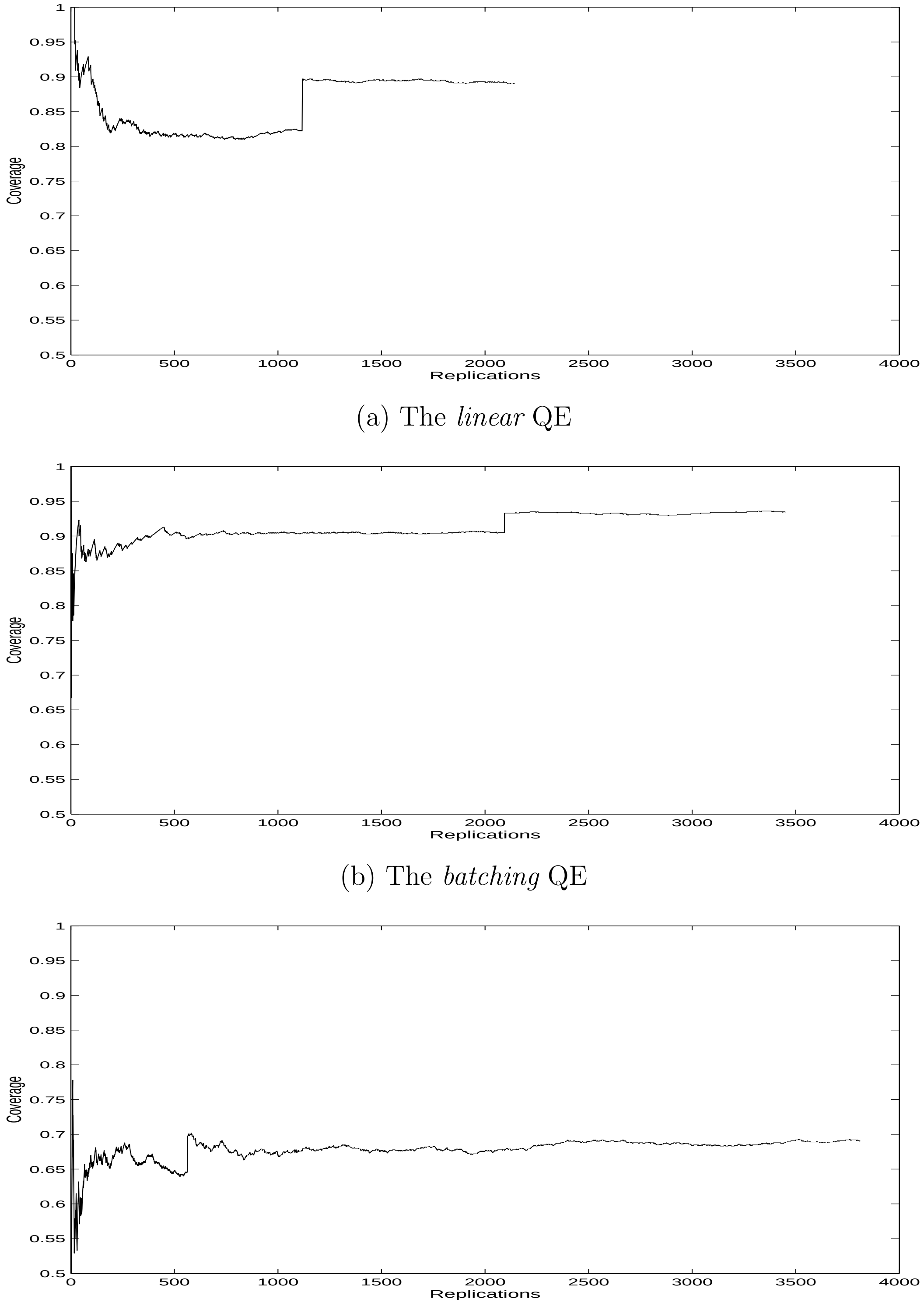


(a) The *linear* QE

(b) The *batching* QE

(c) The *spectral* P^2 QE

Figure 5.7: Convergence of the coverage for QE ($M/M/1/\infty$ at $\rho = 0.5$ and 0.9 quantile, confidence level of 0.95)



(a) The *linear* QE

(b) The *batching* QE

(c) The *spectral* P^2 QE

Figure 5.8: Convergence of the coverage for QE after discarding 'too short' simulation runs ($M/M/1/\infty$ at $\rho = 0.5$ and 0.9 quantile, confidence level of 0.95)

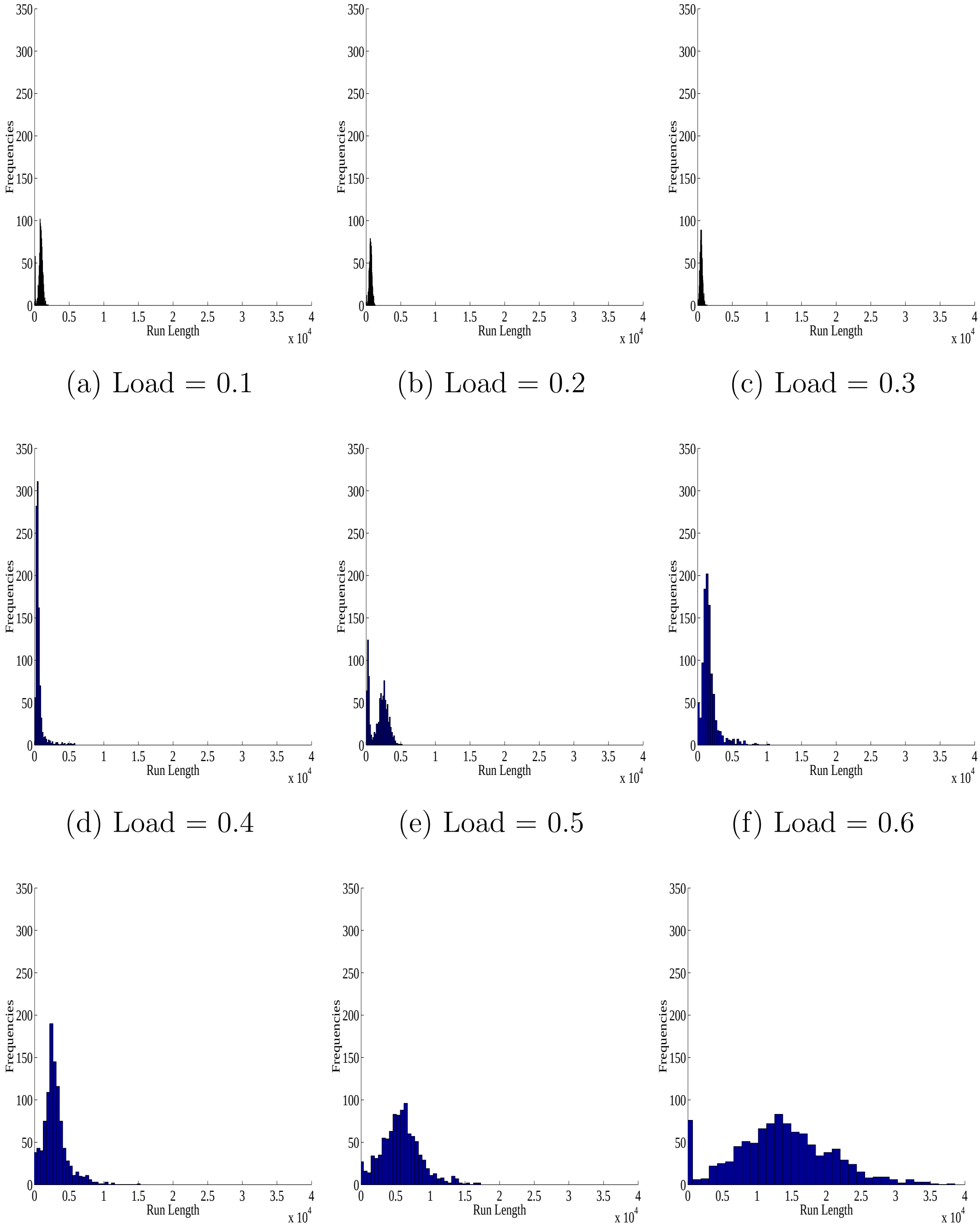
5.5 Performance Evaluation of Sequential QE Approaches

All numerical results refer to a sequential steady-state simulation stopped when the final steady-state estimate of the 0.9 quantile of the response time in the $M/M/1/\infty$ queueing system, selected as a basic model, reached the required relative statistical error of 10% or less, at the 0.95 confidence level. For the *linear* QE, we have assumed two options of 21 grid points spaced 0.2 units apart and 31 grid points spaced 0.1 units apart for all observed observations are between the minimum value of the grid (zero) and the maximum value of the grid (four or three), since the theoretical 0.9 quantiles of the response times in the $M/M/1/\infty$ queueing system are from 0.255843 at $\rho = 0.1$ to 2.302590 at $\rho = 0.9$. (The theoretical 0.9 quantiles can be calculated using Equation (5.12).) We have also assumed the two locations of the first checkpoint[7], to prevent the simulation runs from stopping too early, to be at 100 RCs or more (which is a sufficient number of RCs for the mean value estimation (see Chapter 4) and 1,000 RCs or more (which is assumed for QE in [72]).

To determine whether $n_1 = 100$ RCs is appropriate as the location of the first checkpoint for *linear* QE, we have shown the distribution of run-lengths obtained from 1,000 simulation runs in Figure 5.9. These results clearly show that $n_1 = 100$ RCs is improper, since the phenomenon of 'extremely short' simulation runs produced by choosing the improper location of the first checkpoint; see Chapter 3 and Chapter 4, has appeared as when the mean value in the RCs method was estimated. Therefore, we applied $n_1 = 1,000$ RCs as the location of the first checkpoint. At this point the 'extremely short' runs disappeared.

The results of coverage obtained with the combinations of the above options are depicted in Figure 5.10. To show the relevance of sequential coverage

[7] According to the results presented in Chapters 3 and 4, the poor quality of the final results obtained from the sequential method of RCs is caused by 'extremely short' simulation runs stopped accidentally. This problem has been solved in the case of estimating mean values by choosing the proper location of the first checkpoint.



(a) Load = 0.1

(b) Load = 0.2

(c) Load = 0.3

(d) Load = 0.4

(e) Load = 0.5

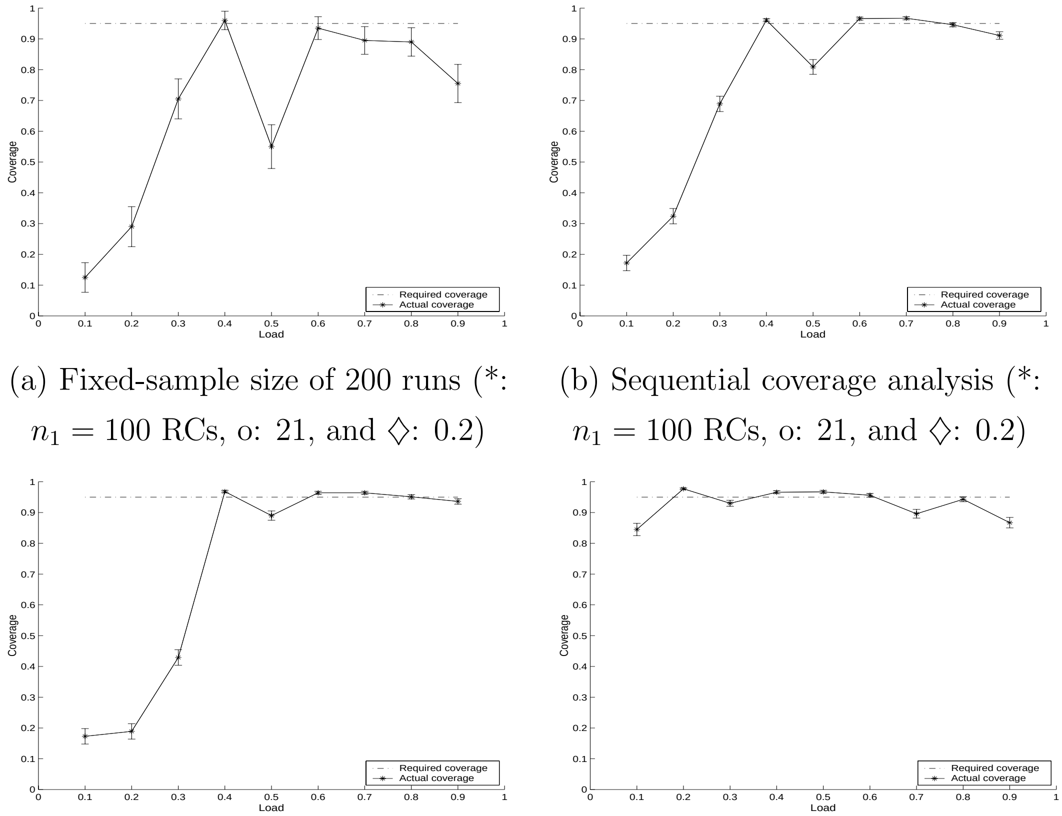
(f) Load = 0.6

(g) Load = 0.7

(h) Load = 0.8

(i) Load = 0.9

Figure 5.9: Simulation run-lengths, measured by the number of RCs, for *linear* QE ($M/M/1/\infty$, 0.9 quantile, the location of the first checkpoint: $n_1 = 100$ RCs, 21 grid points spaced 0.2 units apart)



(a) Fixed-sample size of 200 runs (*: $n_1 = 100$ RCs, o: 21, and ♢: 0.2)

(b) Sequential coverage analysis (*: $n_1 = 100$ RCs, o: 21, and ♢: 0.2)

(c) Sequential coverage analysis (*: $n_1 = 1,000$ RCs, o: 21, and ♢: 0.2)

(d) Sequential coverage analysis (*: $n_1 = 1,000$ RCs, o: 31, and ♢: 0.1)

Figure 5.10: Coverage analysis of the *linear* QE approach in the sequential method of RCs ($M/M/1/\infty$, 0.9 quantile, *: the location of the first checkpoint, o: the number of grid points, and ♢: the space between grid points)

analysis of quantile estimators, we have also obtained the results (based on 200 independent replications[8]) for traditional fixed-sample size analysis of coverage, and depicted them in Figure 5.10. Sequential coverage analysis produces better results than coverage analysis conducted with a fixed-sample size of 200 runs. This is because the final coverage is from the (quite) stable region of coverage and is improved by discarding the 'too short' simulation runs; also see Figures 5.7 and 5.8. In all these cases, simulation runs shorter than a

[8] In simulation literature, many reported results of coverage analysis have usually been obtained from a fixed number of between 10 - 200 replications. As discussed in Chapter 2, such fixed numbers of replications for coverage analysis are not usually sufficient.

threshold (of mean run-length minus one standard deviation of run-lengths) were classified as 'too short'.

However, the coverage analysed with 21 grid points spaced 0.2 units apart is poor especially in lightly loaded traffic whether $n_1 = 100$ RCs or $n_1 = 1,000$ RCs is used as the location of the first checkpoint; see Figure 5.10 (b) and (c). This is caused by the fact that the quantiles obtained are too far away from the theoretical value; see Figure 5.11.

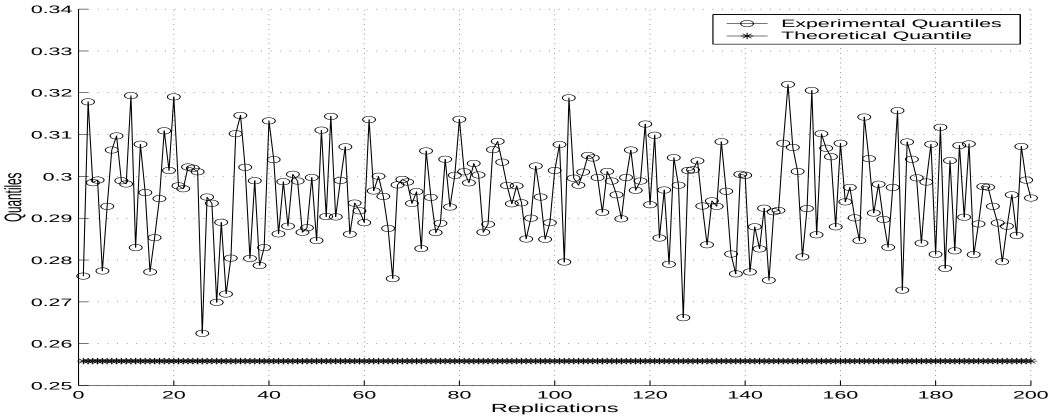


Figure 5.11: *Linear* QE approach in the method of RCs ($M/M/1/\infty$, ρ=0.1, 0.9 quantile, the location of the first checkpoint: $n_1 = 1,000$ RCs, 21 grid points spaced 0.2 units apart)

The coverage analysed with $n_1 = 1,000$ RCs as the location of the first checkpoint and 31 grid points spaced 0.1 units apart has significantly improved especially in lightly loaded traffic; see Figure 5.10 (d). This indicates that the *linear* QE approach is very much affected by the number of grid points and the spacing between them. We can not also guarantee that all bins have sufficient observations to produce reliable quantiles, especially if a particular bin used to calculate a certain quantile may have no observations or fewer observations than other bins. This can definitely produce distinctly biased quantiles. It also requires RCs in a simulated system to be recognised. If a distribution of the observations is known prior, the *linear* QE approach is desirable since the assumptions of the number of grid points and space between them can be optimised. Otherwise, the *linear* QE approach hardly produces reliable quantiles, since these assumptions cannot be optimised.

For the *batching* QE, we have assumed batch sizes, which are the number of RCs per batch, of 100 RCs and 200 RCs, since Seila [163] has recommended using 100 RCs or more to protect against inadequate coverage probabilities. We have also assumed $r_1 = 10$ batches[9] as the location of the first checkpoint. To determine whether the phenomenon of 'extremely short' simulation runs appears when applying this location, we have also depicted the distribution of run-lengths obtained from 1,000 simulation runs in Figure 5.12. These results show that $r_1 = 10$ batches is the proper choice[10].

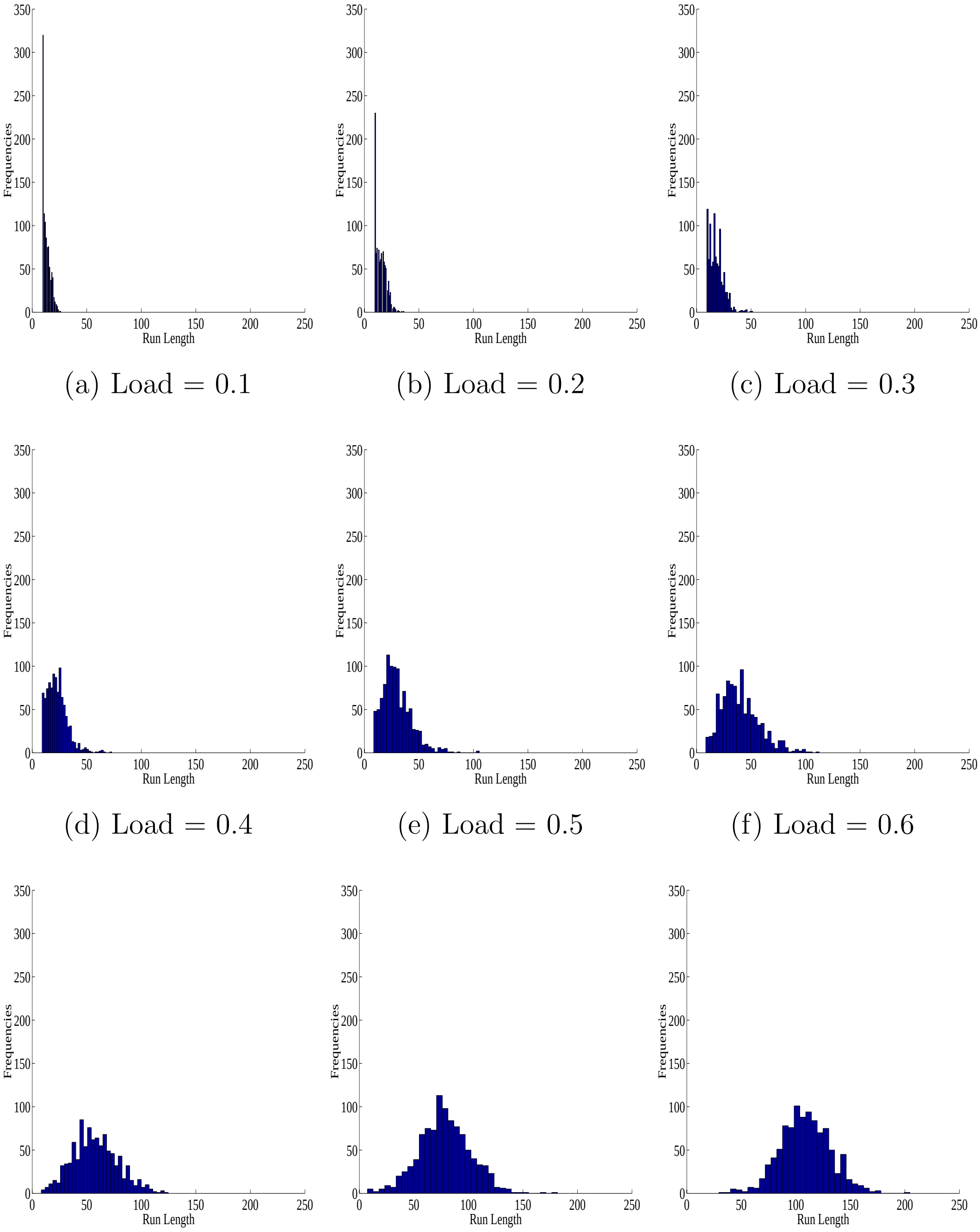
The results of coverage are depicted in Figure 5.13. We have also shown the results obtained for traditional fixed-sample size analysis of coverage and sequential coverage analysis. Sequential coverage analysis produces better results than the traditional coverage analysis as with the *linear* QE approach. The coverage obtained with a batch size of 100 RCs is close to the required level of 0.95, except at a traffic intensity of $\rho = 0.9$.

In the case of $\rho = 0.9$, quantiles are usually underestimated; see Figure 5.14 (depicted only for 200 replications). This can be improved by increasing the batch size. Therefore, we have tested the *batching* QE with a batch size of 200 RCs and depicted the results in Figure 5.13 (c). These results show the significant improvement of coverage especially at a traffic intensity of $\rho = 0.9$. Even though this approach needs some storage for observations collected during 200 RCs and sorting within those collected observations, it is quite desirable, since it is simple and does not need prior knowledge of the distribution of the observations. However, it does need to recognise the RCs in a simulated system.

The results obtained by traditional fixed-sample size analysis of coverage and sequential coverage analysis for the *spectral* P^2 QE approach are depicted

[9] Seila [163] recommended using at least 10 batches. This means the *batching* QE needs to collect at least 1,000 RCs or more.

[10] We have also tested the *batching* QE with $r_1 = 2$ batches as the location of the first checkpoint. That results have shown that this is improper, since 'extremely short' simulation runs appear.

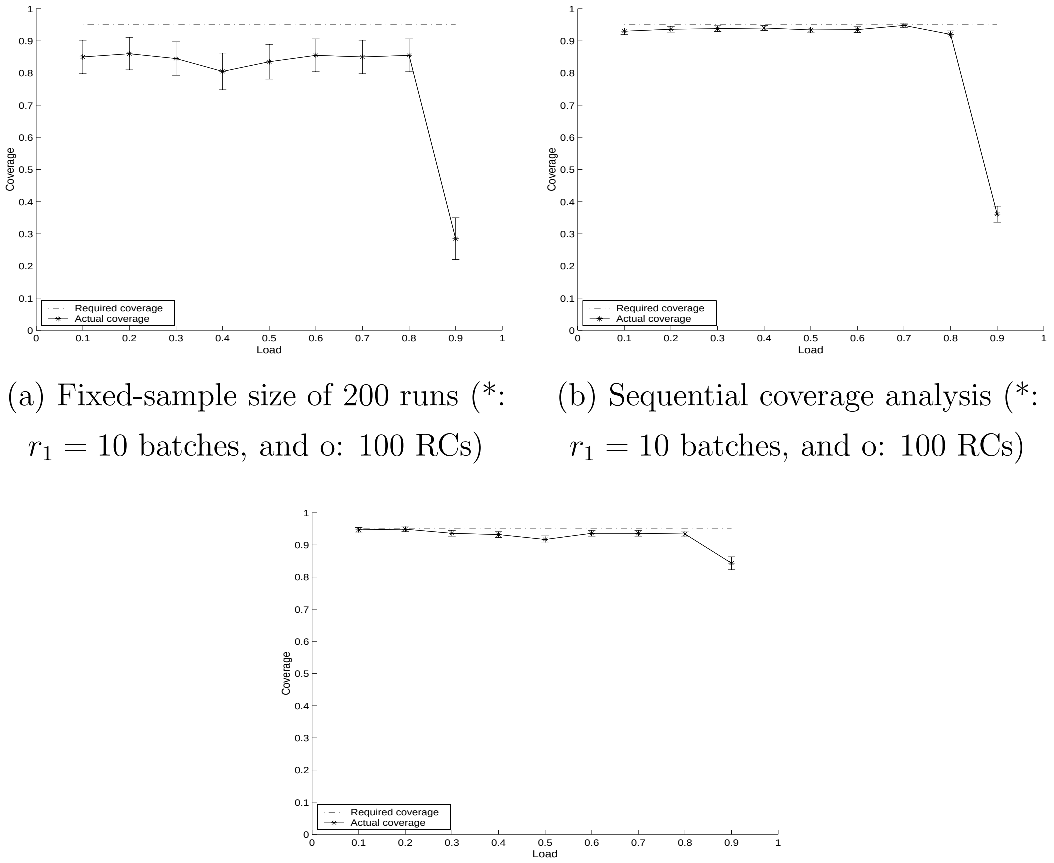


(a) Load = 0.1 (b) Load = 0.2 (c) Load = 0.3

(d) Load = 0.4 (e) Load = 0.5 (f) Load = 0.6

(g) Load = 0.7 (h) Load = 0.8 (i) Load = 0.9

Figure 5.12: Simulation run-lengths, measured by the number of batches, for *batching* QE ($M/M/1/\infty$, 0.9 quantile, the location of the first checkpoint: $r_1 = 10$ batches, the batch size: 100 RCs)



(a) Fixed-sample size of 200 runs (*: $r_1 = 10$ batches, and o: 100 RCs)

(b) Sequential coverage analysis (*: $r_1 = 10$ batches, and o: 100 RCs)

(c) Sequential coverage analysis (*: $r_1 = 10$ batches, and o: 200 RCs)

Figure 5.13: Coverage analysis of the *batching* QE approach in the sequential RCs ($M/M/1/\infty$, 0.9 quantile, *: the location of the first checkpoint, and o: the batch size)

in Figure 5.15. All these results are obtained after discarding observations collected during the initial transient period. Sequential coverage analysis produces better results. However, the *spectral* P^2 QE approach produces poor results especially in terms of coverage, because many numbers of quantiles obtained for the *spectral* P^2 QE are far from the theoretical value; see Figure 5.16.

The mean length of the initial transient periods detected by the Schruben test when estimating 0.9 quantiles using the *spectral* P^2 QE approach from

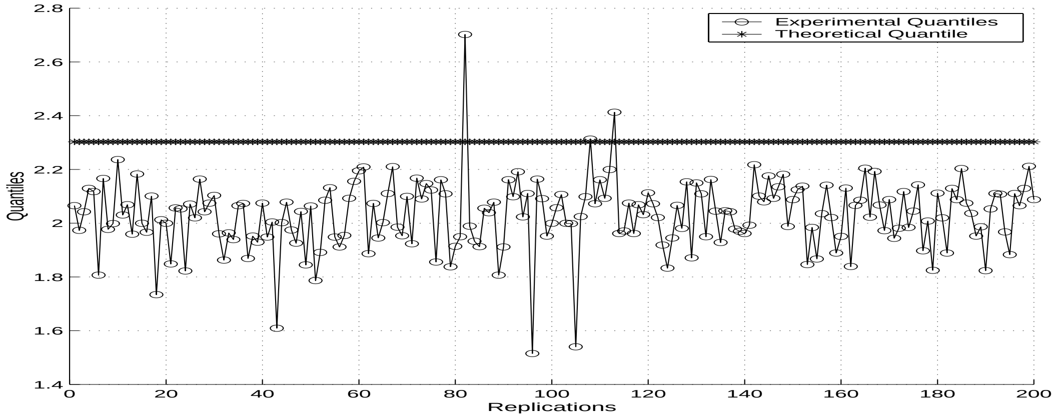
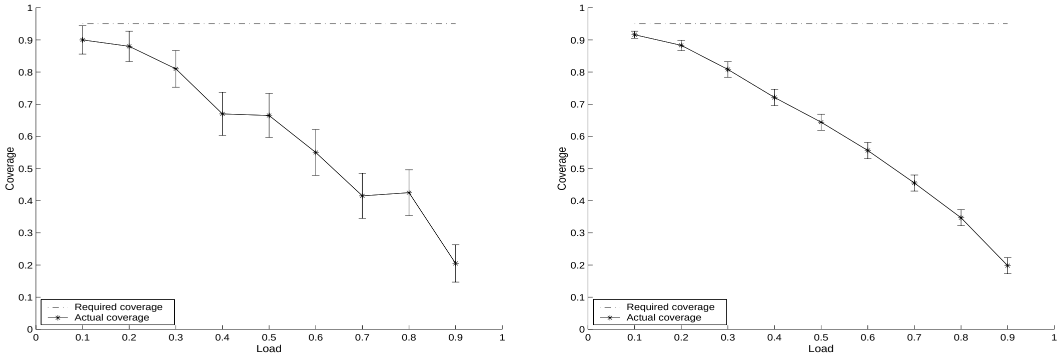


Figure 5.14: *Batching* QE approach in the sequential RCs ($M/M/1/\infty$, ρ=0.9, 0.9 quantile, the location of the first checkpoint: $r_1 = 10$ batches, the batch size: 100 RCs)



(a) Fixed-sample size of 200 runs

(b) Sequential coverage analysis

Figure 5.15: Coverage analysis of the sequential *spectral* P^2 QE approach ($M/M/1/\infty$, 0.9 quantile)

6,000 independent replications for the $M/M/1/\infty$ queueing system is also presented in Table 5.2. The length of the initial transient period detected by Schruben test is much shorter than the one, which is determined by a random number between 1,025 and 2,048 generated using a uniform distribution, used by Raatikainen [142]. However, it is enough to eliminate the initialisation bias without the loss of many good observations.

Table 5.3 shows the means and CIs of quantiles obtained from 6,000 inde-

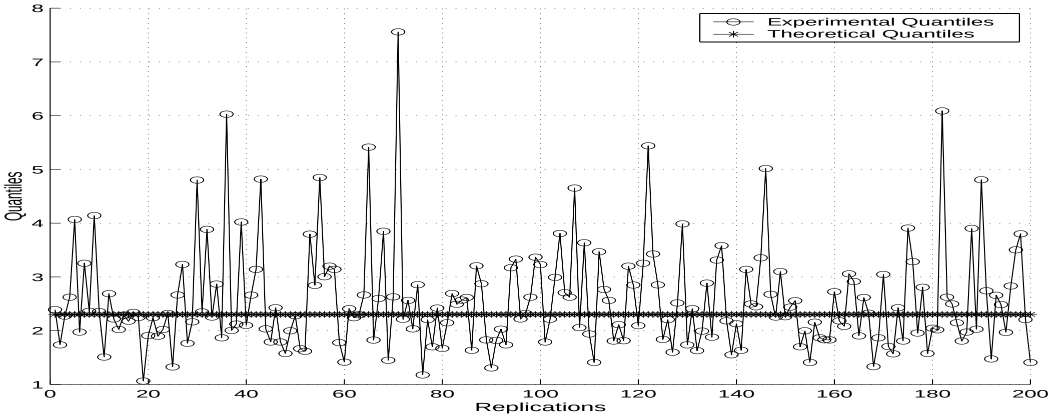


Figure 5.16: *Spectral* P^2 QE approach in the method of non-RCs ($M/M/1/\infty$, ρ=0.9, 0.9 quantile)

Table 5.2: The mean length of the initial transient periods detected by Schruben test when estimating 0.9 quantiles using the *spectral* P^2 QE approach from 6,000 independent replications for the $M/M/1/\infty$ queueing system

Load	*Means of initial transient periods*
0.1	260 ± 7
0.2	267 ± 6
0.3	275 ± 7
0.4	287 ± 7
0.5	302 ± 8
0.6	327 ± 8
0.7	367 ± 9
0.8	441 ± 11
0.9	642 ± 16

Table 5.3: Means and CIs of 0.9 quantiles obtained from 6,000 independent simulation replications executed for the three QE approaches: *linear* QE, *batching* QE, and *spectral* P^2 QE, in the $M/M/1/\infty$ queueing system at a confidence level of 0.95 ($\surd$ means that the CIs of a quantile contain the theoretical quantile)

ρ	*Quantiles in Theory*	*Means & CIs of linear QE*	*Means & CIs of batching QE*	*Means & CIs of spectral*P^2 *QE*
0.1	0.255843	0.266191 [0.259498, 0.272968]	0.255504 $\surd$ [0.249080, 0.262009]	0.256416 $\surd$ [0.249968, 0.262944]
0.2	0.287823	0.291034 $\surd$ [0.283716, 0.298444]	0.286768 $\surd$ [0.279557, 0.294069]	0.289103 $\surd$ [0.281834, 0.296464]
0.3	0.328941	0.335710 $\surd$ [0.327269, 0.344257]	0.326654 $\surd$ [0.318440, 0.334971]	0.332504 $\surd$ [0.324143, 0.340969]
0.4	0.383764	0.384938 $\surd$ [0.375258, 0.394738]	0.380337 $\surd$ [0.370774, 0.390021]	0.390177 $\surd$ [0.380366, 0.400111]
0.5	0.460517	0.463603 $\surd$ [0.451946, 0.475407]	0.455172 $\surd$ [0.443727, 0.466761]	0.471884 $\surd$ [0.460019, 0.483898]
0.6	0.575646	0.572436 $\surd$ [0.558042, 0.587010]	0.567250 $\surd$ [0.552986, 0.581692]	0.598979 [0.583918, 0.614229]
0.7	0.767528	0.758939 $\surd$ [0.739855, 0.778262]	0.755950 $\surd$ [0.736941, 0.775196]	0.805627 [0.785370, 0.826139]
0.8	1.151290	1.133944 $\surd$ [1.105432, 1.162815]	1.135087 $\surd$ [1.106545, 1.163987]	1.23074 [1.202068, 1.264468]
0.9	2.302590	2.056829 [2.005110, 2.109197]	2.184843 [2.129905, 2.240470]	2.520956 [2.457568, 2.585141]

pendent simulation replications executed for the three QE approaches: *linear* QE[11], *batching* QE[12], and *spectral* P^2 QE, in the $M/M/1/\infty$ queueing system. Note that the *spectral* P^2 QE approach produces slightly greater quantiles with

[11] With $n_1 = 1,000$ RCs as the location of the first checkpoint and 31 grid points spaced 0.1 units apart.

[12] With $r_1 = 10$ batches as the location of the first checkpoint and the batch size of 200 RCs.

increasing the traffic intensity. This agrees well with the results obtained using the *extended* P^2 method of Raatikainen [141]. This clearly causes poor coverage in the *spectral* P^2 QE approach. The results presented in Table 5.3 show that the *batching* QE approach is the best in terms of the CIs of quantiles covering the theoretical quantiles.

We have also presented the results of bias obtained for the three QE approaches, from the same data used in Table 5.3, in Figure 5.17. The bias measures the systematic deviation of the estimator from the true value of the estimated parameter [128]; for example, in the case of the quantile estimate $\hat{Q}_p(n)$, the bias is calculated by

$$\mathrm{Bias}[\hat{Q}_p(n)] = \mathrm{E}[\hat{Q}_p(n) - Q_p], \tag{5.24}$$

where Q_p is the theoretical quantile. The results show that the bias becomes severe with increasing traffic intensity. The *batching* QE approach is less biased than the others at a traffic intensity of $\rho = 0.9$.

Comparing the results presented so far, one can see that the best results in the analysis of very dynamic queueing processes can be achieved by applying *batching* QE if one chooses the proper location of the first checkpoint and the batch size, and the RCs in a simulated system are recognised easily. Otherwise, the *spectral* P^2 QE approach, which produces slightly greater quantiles, is an

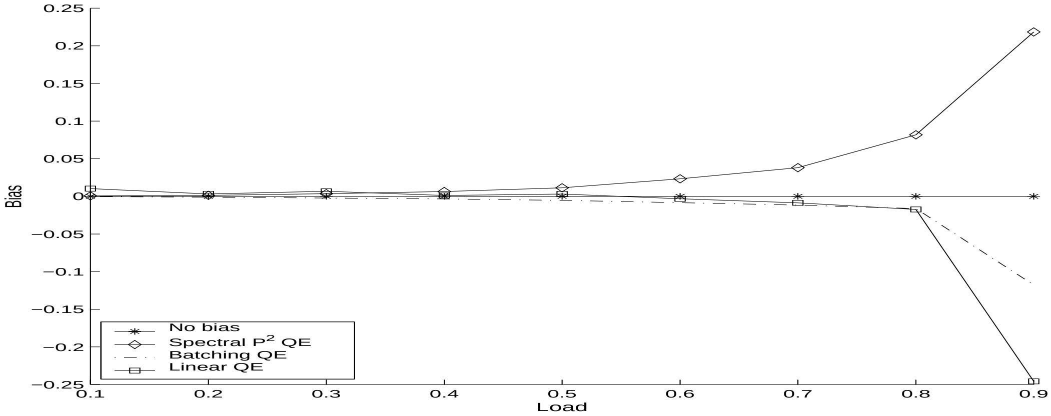
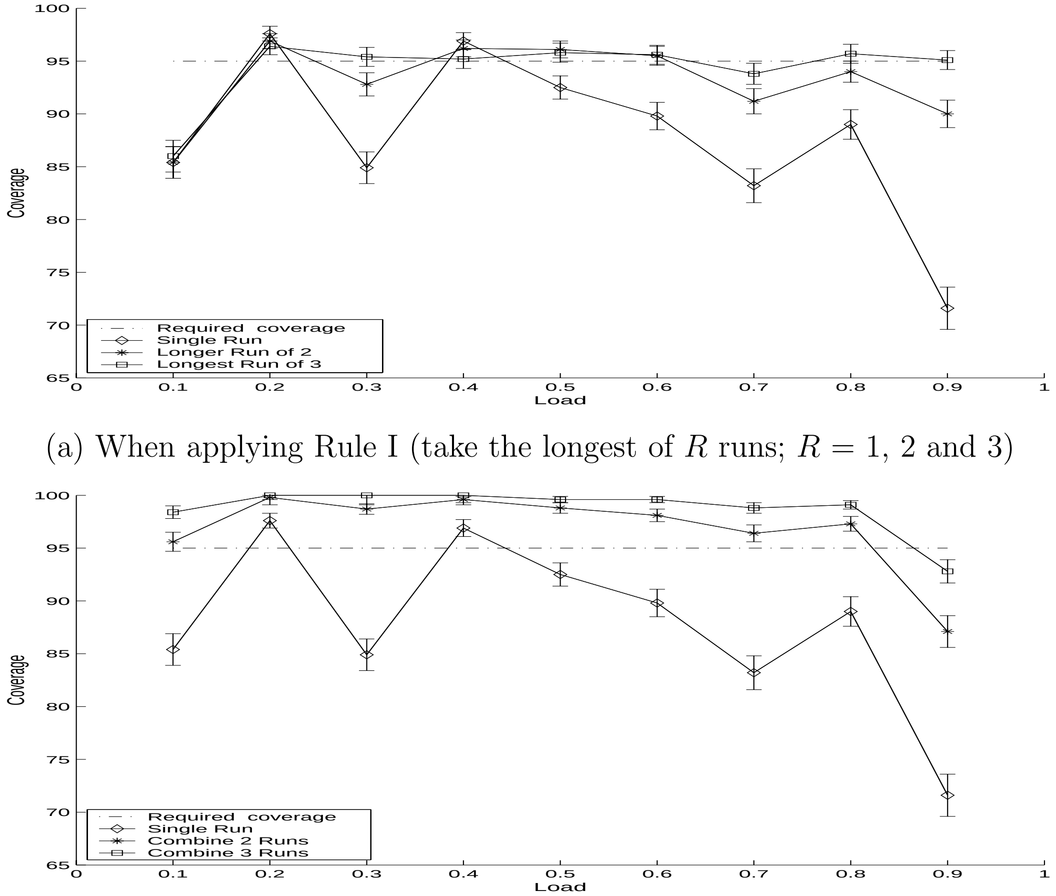


Figure 5.17: Bias of the three QE approaches: the *linear* QE, the *batching* QE, and the *spectral* P^2 QE ($M/M/1/\infty$, 0.9 quantile)

alternative if the small difference is not really important in practice.

We have also applied the two heuristic rules[13]: Rules I and V, which are recommended in Chapter 3 since they can ensure the final results are within an assumed level of confidence or better. The results obtained for each approach are presented in Figures 5.18 - 5.20, respectively.

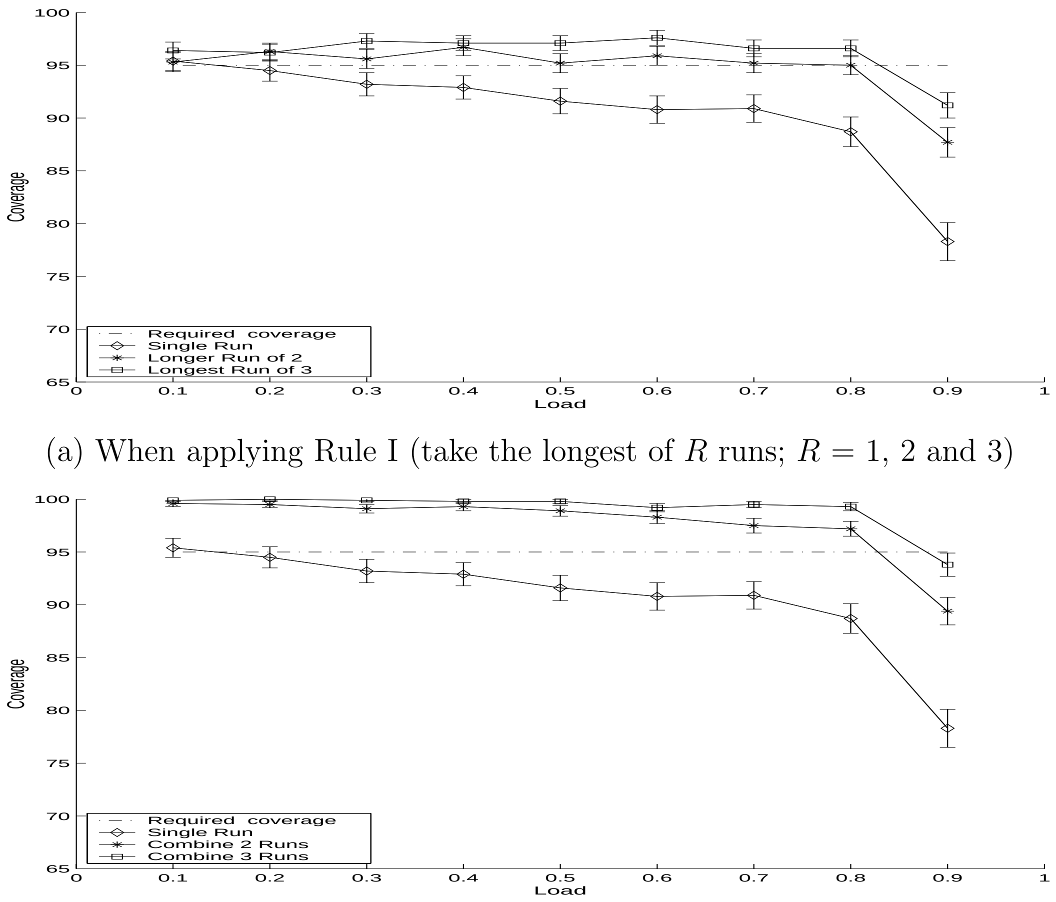


(a) When applying Rule I (take the longest of R runs; $R = 1$, 2 and 3)

(b) When applying Rule V (combination of Rules III and IV)

Figure 5.18: Coverage of the CIs with Rules I and V (proposed in Chapter 3) of the *linear* QE in the sequential method of RCs (when estimating the 0.9 quantile of response times in the $M/M/1/\infty$ queueing system, the location of the first checkpoint: $n_1 = 1,000$ RCs, the number of grid points: 31, and the space between grid points: 0.1)

[13] Proposed for preventing 'too short' runs being included in the final results when estimating the mean value in Chapter 3.

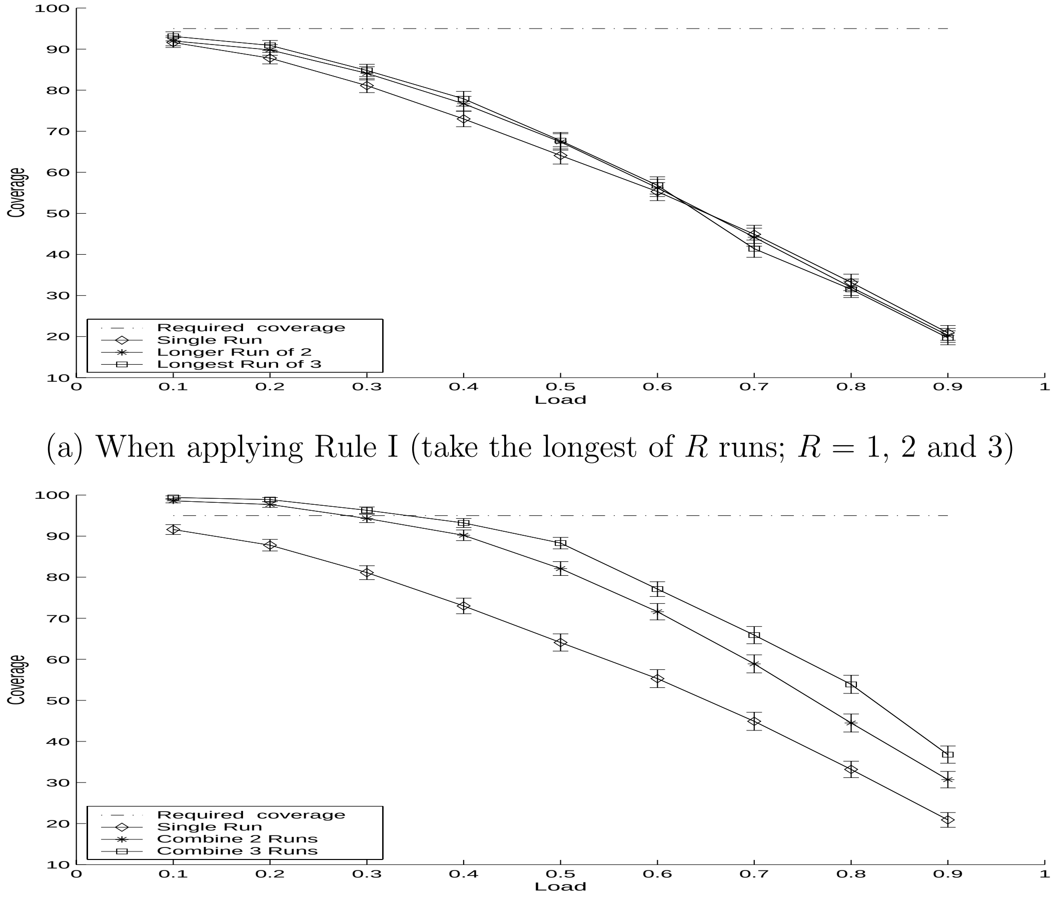


(a) When applying Rule I (take the longest of R runs; $R = 1$, 2 and 3)

(b) When applying Rule V (combination of Rules III and IV)

Figure 5.19: Coverage of the CIs with Rules I and V (proposed in Chapter 3) of the *batching* QE in the sequential method of RCs (when estimating the 0.9 quantile of response times in the $M/M/1/\infty$ queueing system, the location of the first checkpoint: $r_1 = 10$ batches, and the batch size: 200 RCs per batch)

Rules I and V work well for the *linear* QE and *batching* QE approaches. The final results obtained for each approach with Rule I, which is to select the longest run from a few repeated simulation runs, are improved to the (near) assumed confidence level of 0.95; see Figure 5.18 (a) and Figure 5.19 (a). The final results obtained with Rule V are improved over the assumed confidence level of 0.95; see Figure 5.18 (b) and Figure 5.19 (b). These results are in good agreement with the results obtained when estimating the mean value; see Chapter 3.



(a) When applying Rule I (take the longest of R runs; $R = 1$, 2 and 3)

(b) When applying Rule V (combination of Rules III and IV)

Figure 5.20: Coverage of the CIs with Rules I and V (proposed in Chapter 3) of the *spectral* P^2 QE approach in the sequential method of non-RCs (when estimating the 0.9 quantile of response times in the $M/M/1/\infty$ queueing system)

However, Rule I does not work for the *spectral* P^2 QE approach at all; see Figure 5.20 (a), while Rule V slightly improves the final results, but not to a satisfactory level; see Figure 5.20 (b). This is because, as discussed before, quantiles estimated using the *spectral* P^2 QE approach are slightly greater than the theoretical quantiles.

5.6 Conclusions

Quantiles are convenient measures of an entire range of simulation output data. However, the direct estimation of quantiles, based on storing and consecutive multiple sorting of entire sequences of observations collected during the sequential steady-state simulation, appears to be impractical in real applications of a stochastic simulation. In general, the computation time to sort n observations is $O(n \log_2 n)$ and memory proportional to n is required to store sorted values in order to find a given order statistic.

A few quantile estimators, which overcome the inherent limitations of QE, have been proposed so far. Those approaches for estimating quantiles only require linear computation time and little memory. However, most of them are based on the traditional fixed-sample size approach, even though the sequential approach is generally recognised as the more credible approach in controlling the final statistical error in a stochastic simulation.

In this chapter we have studied the properties of three sequential quantile estimators: *linear* QE and *batching* QE for the method of RCs, and *spectral* P^2 QE for the method of non-RCs, to determine the best one to implement in a fully automated simulation package such as Akaroa-2 [28]. As our results show, only the *batching* QE approach offers a reasonable quality of the final results, in terms of coverage analysis and bias, when estimating the response times in the $M/M/1/\infty$ queueing system. However, the *batching* QE approach does require recognition of the RCs in a simulated system. If this is the case then the *batching* QE approach is a good method. Otherwise, the *spectral* P^2 QE approach, which produces the poor coverage caused by slightly greater quantiles, can be an alternative since the poor coverage can be improved by assuming a higher statistical error of the final results.

We have also studied two statistical tests to determine the suitability of applying them in the case of QE: Schruben's test and the GSS test, which were originally developed for detecting the initial transient period when estimating the steady-state mean [162], and [57]. Our results show that these tests also work when estimating the steady-state quantiles. Schruben's test appears to be better, since the empirical quantiles of the waiting time in the queue obtained with it are much closer to their theoretical steady-state quantiles, with smaller errors.

One of the inherent problems in sequential steady-state simulations is that a simulation run can be very short since the stopping condition can be accidentally and temporarily satisfied. These short runs degrade the quality of the final result. This problem has been solved by applying the rules of experimental coverage analysis (discussed in Chapter 2) for the three QE approaches: *linear* QE, *batching* QE, and *spectral* P^2 QE. The other problem, especially in the sequential method of RCs, is that 'extremely short' simulation runs are produced by choosing an improper location of the first checkpoint. This problem also occurred when estimating quantiles using the *linear* QE and *batching* QE approaches based on the method of RCs; similarly for the mean value estimation in the RC method. This problem has been solved by choosing a proper location of the first checkpoint, after collecting at least 1,000 RCs.

The two recommended heuristic rules: Rules I and V, proposed for preventing 'too short' simulation runs being included in the final results (in Chapter 3), have been applied for the three QE approaches. These rules work well for the *linear* QE and *batching* QE approaches, but they do not work for the *spectral* P^2 QE approach since this approach produces slightly greater quantiles.

Chapter 6

SPEEDUP IN PARALLEL AND DISTRIBUTED SIMULATION

6.1 Introduction

Dynamic discrete-event systems, such as manufacturing processes, telecommunication networks, computer systems, etc., are difficult to evaluate analytically due to their complex and often nonlinear behaviour. Simulation is often the only way to evaluate such systems. Some real-world situations, such as controlling air traffic, commanding a large military operation, and determining issues in a competitive marketplace, need to have simulation results in a very short time [41]. However, studies of even moderately complex systems can require excessive computing time to obtain statistically accurate results.

Recent advances in technology, the availability of fast processors, and large memories have helped those computationally intensive simulations. However, dynamic discrete-event simulation still creates a significant computational problem with increasing model complexity. Therefore, parallel and distributed simulation holds great promise for meeting the simulation needs of developers of increasingly complex systems, since the use of parallel and distributed computer systems can significantly speed up run times, enabling a simulation program to execute on a computing system containing multiple processors, such as personal computers interconnected by a communication network.

The required number of observations increases dramatically as the autocorrelations increase. For instance, the required run-lengths to estimate the mean response time in a steady-state simulation of an $M/M/1/\infty$ queueing system with traffic intensities $\rho = 0.9$, $\rho = 0.99$ and $\rho = 0.999$, are 582386, 61156736, and 6143485196, respectively. The procedure of calculating the theoretically required run-length can be found in Appendix F. The number of observations collected is proportional to the CPU time. Our experiments show that the estimation of the mean response time in the $M/M/1/\infty$ queueing system requires roughly 8.3 minutes ($\rho = 0.99$) and 1.3 days ($\rho = 0.999$) on a 350 Mhz Pentium II to achieve an estimate with a relative statistical error of no more than 5%. For a simple open queueing system with traffic intensities $\rho = 0.99$ and $\rho = 0.999$, the times to find a steady-state mean response time are approximately 3 hours and 7.3 days, respectively.

Excessive run times hinder the development and validation of simulation models, and can even preclude some performance evaluation studies. There are two possible solutions to this problem. One is to find more efficient estimators, i.e. estimators that require fewer observations to reach a satisfactory level of statistical error [130]. Another obvious solution is to speed up the simulation by executing it on a multiprocessor or distributed computer system by applying the single replication in parallel (SRIP) scenario or the multiple replications in parallel (MRIP) scenario [136]. Detailed discussion of the SRIP and MRIP scenarios can be found in Chapter 1.

We have only considered the MRIP scenario to speed up the simulation in this chapter, since it is able to offer a speedup proportional to the number of processors involved [26], [136], [151]. For example, a 500 station FDDI (Fiber Distributed Data Interface) token ring simulation required approximately 9.7 hours on a single processor, and as little as 5.5 minutes using all 128 processors of an Intel i860 hypercube under the MRIP scenario [151].

In this chapter, we comment on the speedup obtainable in parallel and distributed discrete-event simulations. Theoretical limitations on the speedup of sequential stochastic simulations under the MRIP scenario are discussed in Section 6.2. In Section 6.3, we present the empirical results of the speedup obtained for the MRIP scenario when estimating mean values and quantiles for simulation output data analysis methods based on RCs (regenerative cycles) and non-RCs.

6.2 Theoretical Speedup in the MRIP Scenario

Following Gunther [58], speedup is commonly associated with a measure of parallel numerical performance, and quantifies the reduction in elapsed time achieved by executing a fixed amount of work on a successively greater number of processors. The simplest way of describing the speedup is depicted in Figure 6.1 as an ideal parallelism [58]. Ideal parallelism assumes that a total simulation time which runs on a uniprocessor in time $T(1)$ can be fully partitioned and executed on P homogeneous simulation processors simultaneously in time $T(1)/P$ [58]. This can give a linear speedup.

However, most simulations cannot be partitioned in this ideal way because some portion of the simulation needs to be executed sequentially. Therefore, that portion can only be executed on a single processor. This simulation can be classified into two portions: one can execute in parallel and the other can only execute sequentially; see Figure 6.2 [58]. In this case, defining the parameter $f < 1$, which is a fraction of the simulation which cannot be parallelised (in the steady-state simulation, this corresponds to the relative length of the initial transient period), the total simulation time by P homogeneous simulation processors $T(P)$ is $f \cdot T(1)$ (for the sequential portion) plus $((1-f) \cdot T(1))/P$ (for the parallel portion). Therefore, we can write the time reduction, when assuming a simulation executes using P homogeneous simulation processors, as:

$$T(P) = f \cdot T(1) + \frac{(1-f) \cdot T(1)}{P}. \tag{6.1}$$

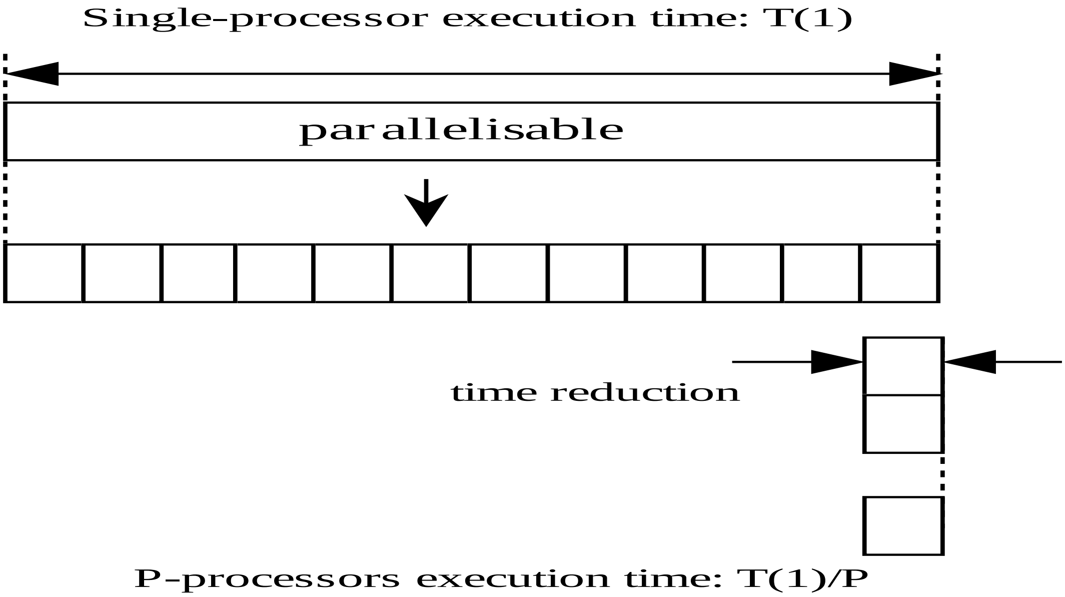

Single-processor execution time: T(1)
parallelisable
time reduction
P-processors execution time: T(1)/P

Figure 6.1: Ideal parallelism (taken from [58])

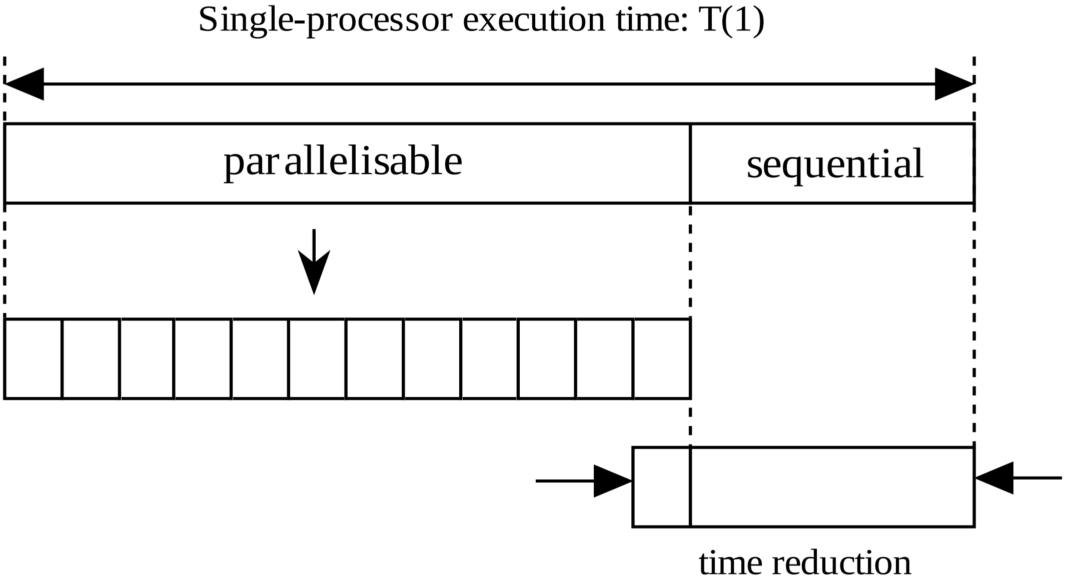


Figure 6.2: Parallelism with two portions: one executes in parallel and the other can only execute sequentially (taken from [58])

The conventional definition of the speedup is given by

$$S(P) = \frac{T(1)}{T(P)}. \tag{6.2}$$

Substituting Equation (6.1) into Equation (6.2), the speedup $S(P)$ is given by

$$S(P) = \frac{T(1)}{\left(f + \frac{1-f}{P}\right) T(1)} \tag{6.3}$$

or

$$S(P) = \frac{P}{1 + f \cdot (P-1)}. \tag{6.4}$$

Speedup achievable with Equation (6.4) based on *Amdahl's law* is depicted in Figure 6.3. As we can see from Figure 6.3, if the value of the parameter f vanishes ($f = 0.0$), then the speedup would follow the ideal linearly increasing trajectory. Otherwise, depending on the value of f, the speedup falls away from the ideal trajectory. As $P \to \infty$, Equation (6.4) has an asymptotic

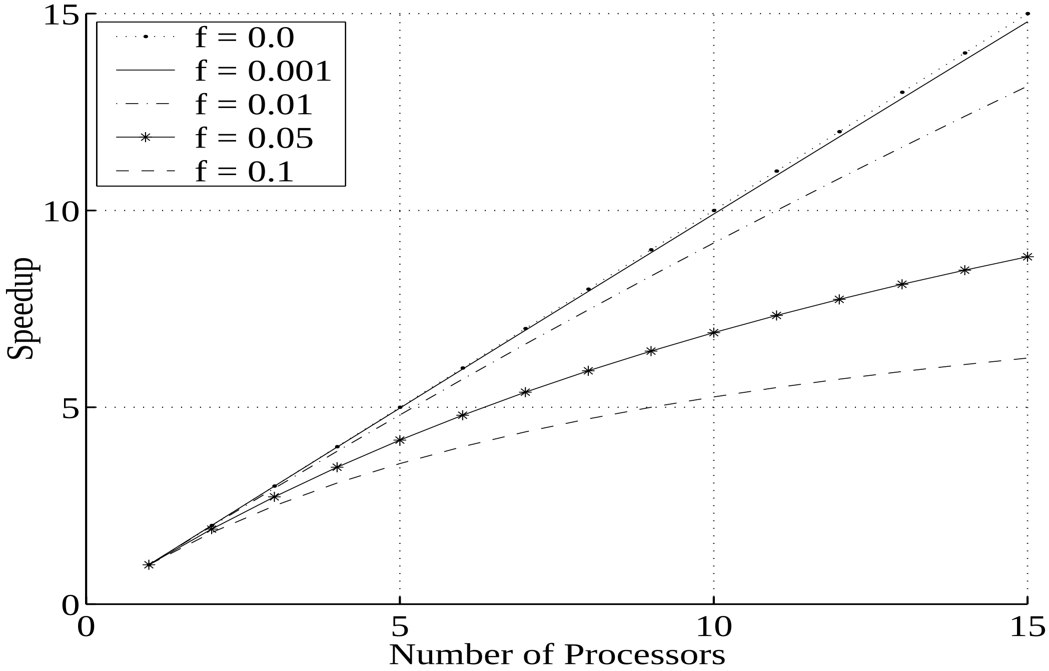


Figure 6.3: Speedup achievable with Equation (6.4) based on *Amdahl's law*

bound at $1/f$. Equation (6.4) can be regarded as an upper limit of the speedup, since it assumes each parallel subtask is homogeneous with identical processing demands [58]. In reality, applications are less uniform. Therefore, the speedup will be inferior to that expected on the basis of *Amdahl's law* [26], [133].

Following [133], to analyse the average speedup of sequential steady-state simulation runs under the MRIP scenario, let us note that each processor runs an independent replication of the simulation process. Therefore, it first generates data characterising the initial transient period (if there is such a period) and these data are discarded. Only later, having entered the steady-state region, does a simulation processor start its contribution to the steady-state analysis by submitting its data to a global analyser.

Obviously, the best speedup is achievable if one launches simulation processors on an homogeneous set of processors. With heterogeneous processors speeding up the simulation may not even be possible. This case occurs when one of the processors is fast enough to generate the required number of observations before any of the slower processors reaches the first checkpoint. Therefore, we assume that a steady-state simulation is run on a set of P homogeneous simulation processors, and the length of the simulation is measured by the total number of observations submitted by P simulation processors to the global analyser before the simulation is stopped. Furthermore, assuming the very fine granularity of a stochastic simulation (the small distance between checkpoints), the speedup of a steady-state simulation in the MRIP scenario would be governed by the *truncated Amdahl's law* [133].

Following [133], let us assume that a sequential steady-state simulation under the MRIP scenario is stopped when P homogeneous simulation processors have delivered the total number of observations N_{min} needed to satisfy the stopping criterion. As the number of processors increases, we will reach a situation in which all P processors reach their first checkpoint before the global analyser stops the simulation. Let D be the location of the first checkpoint (i.e., the number of observations generated when the first checkpoint is reached), and let

$$P_{min} = min\{P : D \cdot P \geq N_{min}\}. \tag{6.5}$$

Adding more than P_{min} processors would not increase the speedup, since it has already reached its maximum speedup of

$$S_{max} = P_{min}. \tag{6.6}$$

The effect of adding more than P_{min} processors is that the total number of observations when the simulation is stopped is greater than N_{min}. Having more observations (generated by $P > P_{min}$ processors) only improves the statistical error. Therefore, the upper limit of the maximum speedup can be rewritten as

$$S_{max} = \frac{N_{min}}{D}. \tag{6.7}$$

Linking Equations (6.3) and (6.7) leads to the following *truncated Amdahl's law* proposed by Pawlikowski and McNickle in [133]:

$$Sp(P) = \begin{cases} \frac{1}{f+(1-f)/P} & \text{for } P < P_{min} = \frac{(1-f)N_{min}}{D}, \\ \\ \frac{1}{f+D/N_{min}} & \text{for } P \geq P_{min} = \frac{(1-f)N_{min}}{D}, \end{cases} \tag{6.8}$$

where f is the relative length of the initial transient period, which means the simulation cannot be parallelised, P is the number of processors ($P > 1$), D is the location of the first checkpoint, N_{min} is the total number of observations needed, and $Sp(P)$ is the speedup achievable with P homogeneous simulation processors. The speedup obtained from the *truncated Amdahl's law* of Equation (6.8), when assuming $P_{min} = 10$ processors, is plotted in Figure 6.4.

As discussed in [133], one can draw the following conclusions from these results:

- To obtain maximum speedup under the MRIP scenario, P_{min} processors or more are needed to collect the required number of observations.
- The longer the relative length of the initial transient period, the smaller the speedup. As the value of parameter f increases, the speedup falls away from the theoretical trajectory.
- The *truncated Amdahl's law* is valid for average speedup.

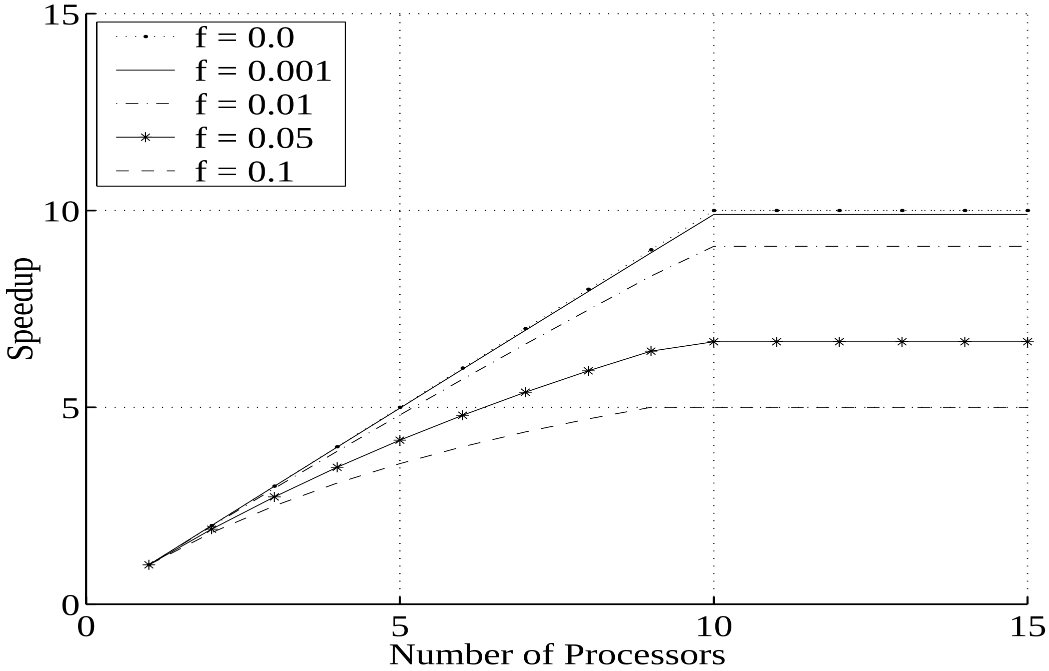


Figure 6.4: Speedup achievable theoretically under the MRIP scenario according to the truncated Amdahl's law ($P_{min} = 10$)

- If the length of the initial transient period is negligible in comparison with the total simulation run-length, or the length of the initial transient period has no role in the steady-state analysis, such as in the method of RCs, then the speedup should be linear with the number of processors engaged.

6.3 Empirical Speedup in the MRIP Scenario

To analyse the speedup of parallel and distributed simulations under the MRIP scenario, one needs to take into account specific computational requirements of the method used to estimate the variance and its statistical error. Speedup obtained in the MRIP scenario has been reported in [26], [131], and [136]. Although, potentially, speedup improves in proportion to the number of processors involved, in practice it can depend heavily on the method used to estimate the variance. Ewing et al. [26] reported that the speedup obtained using the SA/HW method is close to the expected theoretical value and much better than using NOBM, since in the SA/HW method each processor begins sending estimates to the global analyser after the initial transient period is over without other startup overheads.

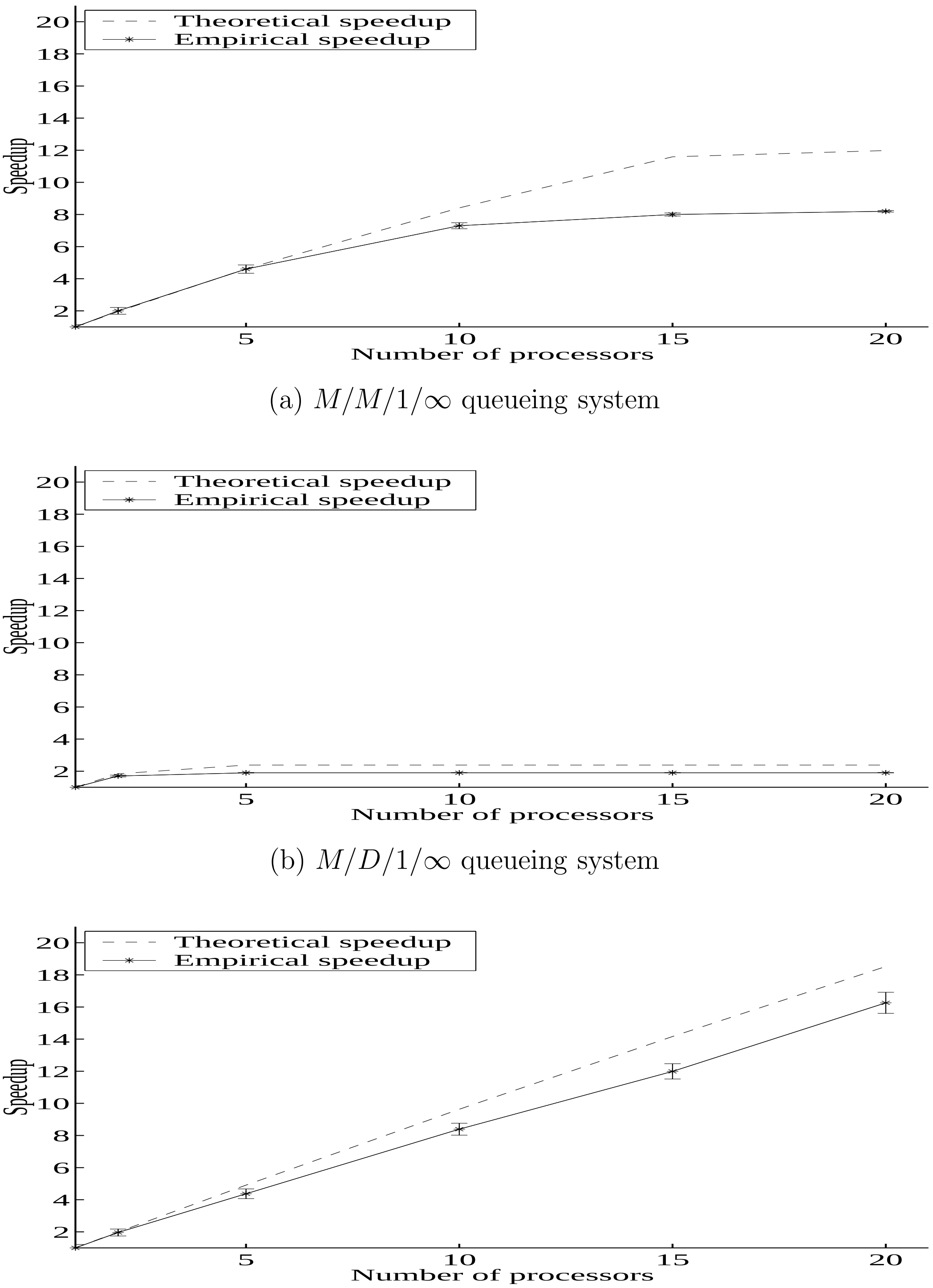
Empirical results of the speedup obtained under the MRIP scenario, applying the SA/HW method to analyse a steady-state estimate's variance and mean response time in the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems at a traffic intensity of $\rho = 0.5$ with a relative statistical error of at least 5% at the 0.95 confidence level, are depicted in Figure 6.5. All empirical results are averaged from 100 independent sequential steady-state simulation runs executed using Akaroa-2 [28]. Statistics averaged from these runs are also presented in Table 6.1.

Table 6.1: Statistics averaged from 100 independent sequential steady-state simulation runs applying the SA/HW method to estimate the mean response time at a traffic intensity of $\rho = 0.5$ with a relative statistical error of at least 5% at the 0.95 confidence level

	$M/M/1/\infty$	$M/D/1/\infty$	$M/H_2/1/\infty$
Total Length	14833 ± 3045	3289 ± 676	110576 ± 22700
Initial Length	309 ± 63	282 ± 58	465 ± 96
Initial/Total (f)	2.08 ± 0.56%	8.57 ± 2.31%	0.42 ± 0.11%
No. of Checkpoints	15.17 ± 3.11	2.9 ± 0.59	82.83 ± 17

When sequential steady-state simulations were executed using Akaroa-2 [28], we found similar results to those reported in [26] for the $M/M/1/\infty$ and $M/H_2/1/\infty$ queueing systems. Particularly, the speedup for the $M/H_2/1/\infty$ queueing system is almost linear. This is not surprising since the total length of the simulation is quite long and the relative length of the initial transient period is quite short.

The speedup obtained for the $M/D/1/\infty$ queueing system is not significant because the required number of observations to meet the stopping criteria



(a) $M/M/1/\infty$ queueing system

(b) $M/D/1/\infty$ queueing system

(c) $M/H_2/1/\infty$ queueing system

Figure 6.5: Speedup obtained under the MRIP scenario for the SA/HW method when estimating steady-state mean response times ($\rho = 0.5$)

are usually collected after only three checkpoints. The theoretically required number of observations for the $M/D/1/\infty$ queueing system is quite small compared to the other systems; see Appendix F for the theoretically required number of observations, and Table 6.1 for the empirically collected number of observations. Therefore, the speedup can only reach a low level of the threshold when the relative length of the initial transient period (f) is also considered. The speedup obtained from SA/HW, in general, follows the *truncated Amdahl's law* closely.

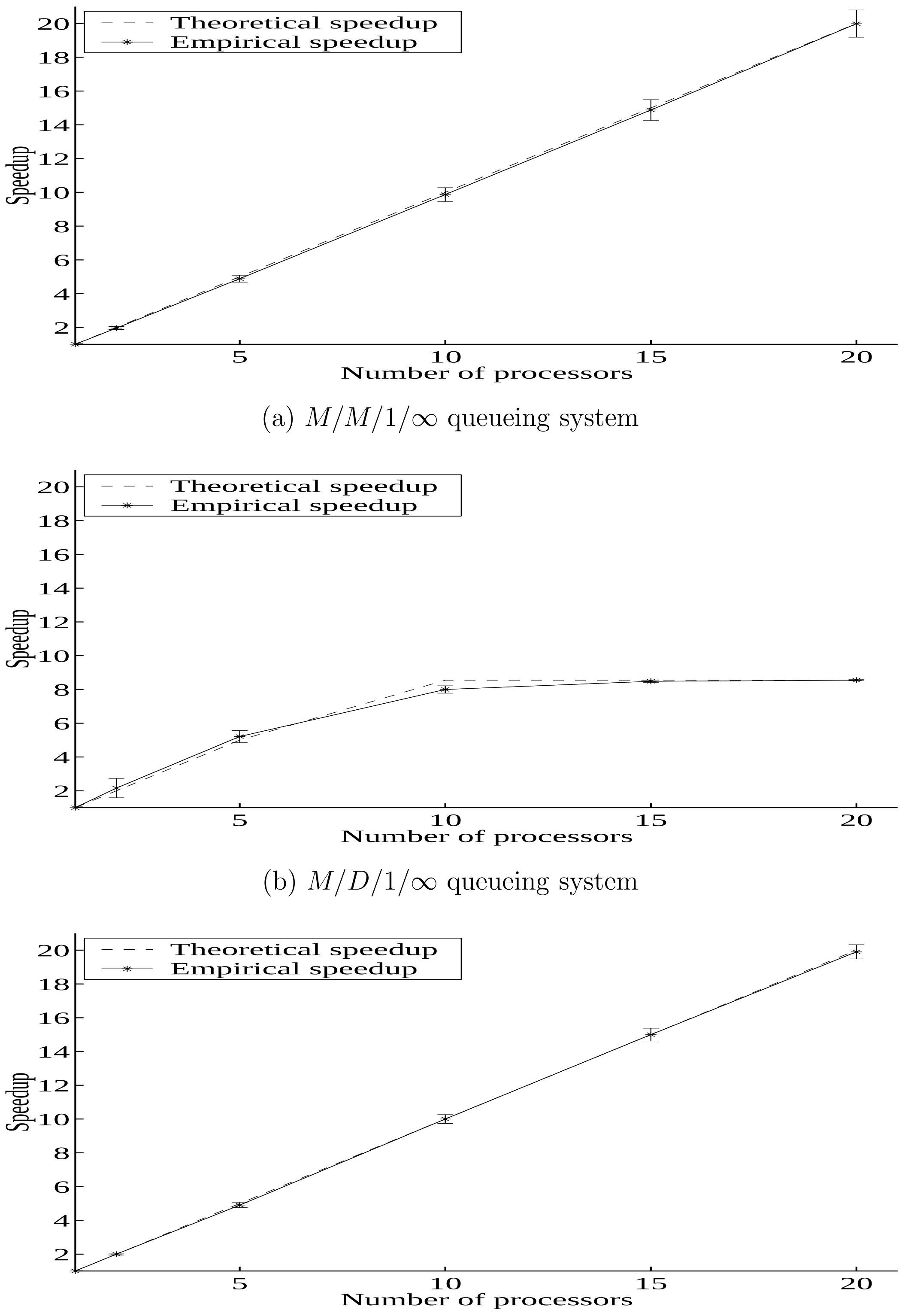
As shown in Figures 6.3 and 6.4, the speedup is very much affected by the relative length of the initial transient period. However, the RC method under the MRIP scenario should achieve a linear speedup proportional to the number of processors engaged, since this method has no problem with the initial transient period. Empirical results of the speedup obtained under the MRIP scenario, applying the RC method[1] to analyse a steady-state estimate's variance and mean response time in the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems at a traffic intensity $\rho = 0.5$ with a relative statistical error of at least 5% at the 0.95 confidence level, are depicted in Figure 6.6.

As expected, the speedup for the $M/M/1/\infty$ and $M/H_2/1/\infty$ queueing systems is almost linear to the number of processors engaged since the run-lengths of the simulation are quite long. However, the speedup in the $M/D/1/\infty$ queueing system has not increased linearly because the total number of observations needed to satisfy the stopping criterion is collected at the first checkpoint when using about nine processors; see Table 6.2[2]. Note that the location of the first checkpoint is 100 RCs ($n_1 = 100$). The numbers of observations and RCs collected by a single processor in the $M/D/1/\infty$ queueing system are relatively smaller than the other systems. As discussed before, adding more than nine processors in the case of the $M/D/1/\infty$ queueing system does not improve the speedup but increases the total number of observations. These results agree well with the *truncated Amdahl's law.*

The *truncated Amdahl's law* is also applicable to the estimation of steady-

[1] The location of the first checkpoint is 100 RCs ($n_1 = 100$).

[2] See Appendix F for the theoretically required number of observations.



(a) $M/M/1/\infty$ queueing system

(b) $M/D/1/\infty$ queueing system

(c) $M/H_2/1/\infty$ queueing system

Figure 6.6: Speedup obtained under the MRIP scenario for the method of RCs when estimating steady-state mean response times ($\rho = 0.5$)

Table 6.2: Statistics averaged from 100 independent sequential steady-state simulation runs applying the RCs method to estimate the mean response time at a traffic intensity $\rho = 0.5$ with a relative statistical error of at least 5% at the 0.95 confidence level

	$M/M/1/\infty$	$M/D/1/\infty$	$M/H_2/1/\infty$
Theoretically Required Obs.	16903	3073	131385
Num. of Obs. Collected by P=1 (Proportion)	13015 ± 2715 (76.99 ± 16.07%)	1704 ± 350 (55.45 ± 11.39%)	119025± 24435 (90.59 ± 18.60%)
Num. of RCs Collected by P=1	6629 ± 2048	855 ± 176	58961 ± 12104

state quantiles since they can be estimated under the MRIP scenario. Therefore, we have investigated how much speedup can be obtained when estimating steady-state quantiles with the two sequential QE methods: *spectral* P^2 QE (based on the method of non-RCs) and *batching* QE (based on the method of RCs); see Chapter 5 for further discussion of these methods.

Empirical results of the speedup obtained when applying *spectral* P^2 QE to estimate the 0.9 quantile for the response time in the $M/M/1/\infty$ queueing system at a traffic intensity of $\rho = 0.5$ with a relative statistical error of at least 5% at the 0.95 confidence level, are depicted in Figure 6.7. The statistics are also presented in Table 6.3. All empirical results are averaged from 100 independent sequential steady-state simulation runs executed using Akaroa-2 [28]. Figure 6.7 shows that the speedup for estimating quantiles under the MRIP scenario is similar to that of estimating mean values. We can clearly see that the speedup depends on the total run-length of the simulation, the relative length of the initial transient period, and the number of checkpoints observed when running a simulation on a single processor. The empirical speedups have not reached the theoretical speedups. This is because the total number of observations collected has been increased a little excessively by increasing the number of processors.

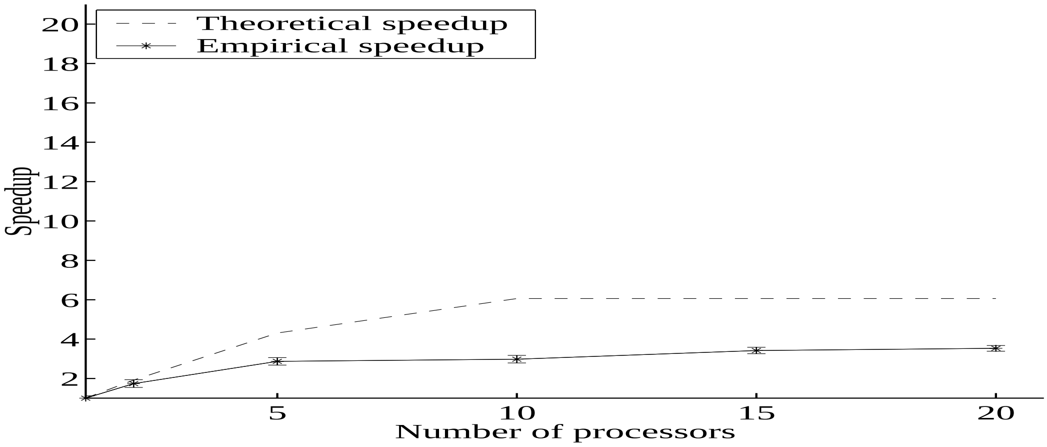


Figure 6.7: Speedup achieved from *spectral* P^2 QE based on the method of non-RCs to estimate the 0.9 quantile for the response time in the $M/M/1/\infty$ at a traffic intensity of $\rho = 0.5$

Table 6.3: Statistics averaged from 100 independent sequential steady-state simulation runs applying the method of *spectral* P^2 QE to estimate the 0.9 quantile for the response time in the $M/M/1/\infty$ at a traffic intensity of $\rho = 0.5$

	$M/M/1/\infty$
Total Length	7631 ± 1567
Initial Length	306 ± 63
Initial/Total (f)	4.02 ± 1.08%
Number of Checkpoints	7.4 ± 1.5

Empirical results of the speedup obtained by applying the method of *batching* QE, based on RCs under the MRIP scenario, to estimate the 0.9 quantile for the response time in the $M/M/1/\infty$ queueing system at a traffic intensity $\rho = 0.5$ with a relative statistical error of at least 5% at the 0.95 confidence level, are depicted in Figure 6.8. All empirical results are also averaged from 100 independent sequential steady-state simulation runs executed using Akaroa-2 [28]. The results show that the speedup is achieved linearly up to with about thirteen processors engaged. Adding more than thirteen processors does increase the total number of collected RCs, but it does not improve the speedup. This is not surprising since the mean of empirically collected RCs is 12670 ± 2601 RCs and the assumed location of the first checkpoint[3] is 1,000 RCs ($n_1 = 1,000$). These results also follow the *truncated Amdahl's law* well.

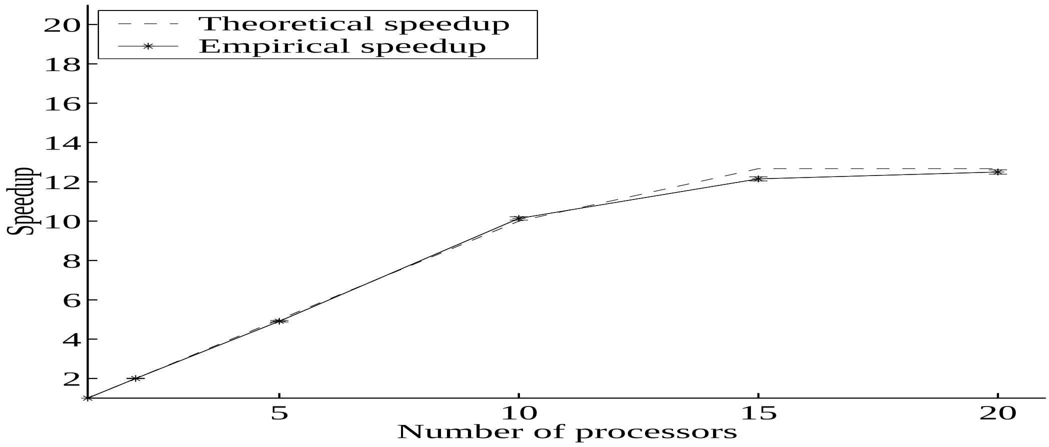


Figure 6.8: Speedup achieved from *batching* QE based on the method of RCs to estimate the 0.9 quantile for the response time in the $M/M/1/\infty$ at a traffic intensity of $\rho = 0.5$

These results lead us to conclude:

- If the distance between checkpoints is too short, say for instance, equals one observation, a simulation running on a single processor takes much longer to collect the required number of observations since the processor has to check the stopping criteria when every observation is generated. However, if these simulations are run under the MRIP scenario, one can offer significant speedup proportional to the number of processors engaged.

[3] $n_1 = 1,000$ RCs as the location of the first checkpoint for QE has been recommended to avoid producing 'extremely short' simulation runs; see Chapter 5.

- The speedups achieved when using the method of RCs to estimate mean values and quantiles differ from theoretically obtainable ones because of the existing granularity of the analysis (significant distance between checkpoints). If the simulation run-length is very long and the number of checkpoint is large, the linear speedup can be achieved because the relative length of the initial transient period is irrelevant and the location of the first checkpoint has also no significant role.

6.4 Conclusions

Although parallel and distributed simulations have led to many important results in different domains, a robust and effective general methodology for dealing with various complex models has not yet been produced. Parallel and distributed simulations can offer especially significant speedup over a sequential steady-state simulation. The effectiveness of the SRIP scenario depends on the level of inherent parallelism in the system to be simulated. If this level is high and the synchronisation, deadlocks and causality errors are properly solved, then the SRIP scenario can significantly speed up the simulation. However, the SRIP scenario has its specific problems and limitations, such as a lack of fault-tolerance, most of which do not occur in the MRIP scenario. The MRIP scenario is almost universally applicable, and is also statistically more efficient in the sense of the mean squared error of the final estimates. The MRIP scenario potentially offers linear speedup proportional to the number of processors involved.

We have commented on parallel and distributed simulations based on the MRIP scenario in terms of the speedup achievable when estimating mean values and quantiles using the methods of RCs and non-RCs. Empirical results obtained using Akaroa-2 show quite good agreement with the *truncated Amdahl's law*, which was derived for estimating the theoretical speedup obtainable under the MRIP scenario in [133], for the speedup of steady-state simulations. The optimal speedup under the MRIP scenario when estimating mean values and quantiles can be achieved if the total run-length of the simulation is very long, the relative length of the initial transient period is very small or zero, and the distance between checkpoints is short. The speedup when using the method of RCs to estimate mean values and quantiles is not affected by the relative length of the initial transient period, but it can be limited by the location of the first checkpoint.

Chapter 7

SUMMARY, CONCLUSIONS, AND FUTURE RESEARCH

In this thesis, we have investigated research issues related to a steady-state simulation. In particular, we have concentrated on how to obtain more credible results using a fully automated simulation tool in both distributed and non-distributed stochastic simulations. A complete summary and conclusions of the main contributions of this thesis follow in Section 7.1. Recommendations for future research are presented in Section 7.2.

7.1 Summary and Conclusions

In Chapter 2, we studied three interval estimators of proportions based on the normal distribution, the *arcsin* transformation and the F distribution, in the context of their application in sequential coverage analysis. Experimental studies of coverage analysis were required to assess the quality of the practical implementations of the methods of simulation output data analysis used to determine CIs in sequential stochastic simulations. Interval estimators for proportions using the (symmetric) normal approximation have been commonly used for coverage analysis of simulation output data even though alternative estimators of (asymmetric) CIs for proportions have been proposed in the past. This is probably because the normal approximation has been easier to use in simulation practice than the other interval estimators. However, current computing technology can now deal with alternative estimators. Even CIs for coverage analysis based on the F distribution can be calculated easily by a standard computer.

Three interval estimators were applied to sequential coverage analysis of the SA/HW method of analysis of steady-state mean response times, in simulations of the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems. Although the numerical results of coverage analysis show that they are very similar in terms of CIs, there are some concerns about their validity. Estimators based on the F distribution have been found to be more accurate and appropriate for use in coverage studies, especially if a higher confidence level is assumed.

In Chapter 2, based on our experimental studies, we also extended some basic rules, proposed originally in [134] for the proper experimental coverage analysis of sequential steady-state simulations. The numerical results of sequential coverage analysis by applying these revised rules were presented for three output data analysis methods: NOBM, SA/HW, and RCs, in simulations of the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems. With those results, we reach better conclusion, since the final CIs at different traffic levels were exactly the same width, unlike the final CIs at different traffic levels obtained with the rules proposed in [134].

In Chapter 3, we addressed the problem of sequential steady-state simulations conducted to study long run mean values of performance measures of stable dynamic systems. The problem is that the inherently random nature of output data collected during the stochastic simulation, due to the pseudo-random nature of input data, can cause an accidental, temporary satisfaction of the stopping rule of such a sequential estimation. It is quite frequently associated with producing a 'too short' simulation run having poor coverage. Our experimental evidence showed that this phenomenon occurs frequently, with a resulting significant degradation of the coverage of the final results.

At the least, lowering the probability of using results from 'too short' simulation runs is one of the very few possible practical ways available for simulation practitioners to improve the quality of the final results from their simulation experiments. We proposed a few simple heuristic rules of thumb that, if applied in practice, can reduce to a negligible level the probability that results come from prematurely finished simulations. The effectiveness of these rules was quantitatively assessed using the results of coverage analysis of the three different methods of simulation output data analysis: NOBM, SA/HW, and RCs, in sequential steady-state simulations. In particular, Rules I and V appear to be effective. However, no rules can guarantee that the final CIs from the sequential stochastic simulation will exactly contain the theoretical value with a probability equal to the assumed confidence level.

To obtain the coverage with an assumed level of confidence, one needs to collect the number of theoretically required observations. However, none of the three methods of simulation output data analysis: NOBM, SA/HW, and RCs, in sequential steady-state simulations for the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems collects the theoretically required number of observations, especially in the case of heavily loaded queueing systems. With the current state-of-the-art simulation output data analysis methods, we cannot assume that the exact level of coverage will be reached. However, we can at least improve the coverage by applying one of Rules I and V, since the number of observations approaches the theoretically required number.

The selection of the appropriate rule depends on the confidence level required. Rule I, which selects the longest run from a few repeated simulation runs, appears to be the most effective in the case where one always wishes to have the final results within an assumed level of confidence, because the coverage from the selected run can be improved to the assumed level of confidence by adjusting the number of replications R. Otherwise, in the case where one always wants to guarantee the final results having a high confidence level, the alternative is Rule V, as the resulting coverage is always between the assumed level of confidence and the maximum level for lightly or heavily loaded queueing systems.

Many methods of steady-state simulation output data analysis, such as spectral analysis, batch means, regenerative cycles (RCs), and standardised time series, have been proposed. However, only the method of RCs can naturally avoid the problem of the initial transient period. Therefore, the method of RCs is a very attractive alternative. However, a sequential steady-state simulation with the RC method can lead to inaccurate simulation results if the simulation experiment stops too early when the sequential stopping criterion is accidentally temporarily satisfied, as can happen in practice from time to time. We investigated this aspect of sequential steady-state simulation with the RC method in Chapter 4.

A problem is that many simulation runs in the sequential method of RCs have often as few as two RCs. This seriously degrades the quality in terms of coverage, unlike the other analysis methods: NOBM and SA/HW. However, if the location of the first checkpoint for sequential RCs is carefully selected, it is possible to avoid collecting those 'extremely short' simulation runs. To observe this, we studied the sequential method of RCs with a number of different locations for the first checkpoint (locating it between $n_1 = 2$ to $n_1 = 150$ RCs). The experimental results clearly showed that the number of 'extremely short' simulation runs is diminished by delaying the first checkpoint. As assuming a minimum number of 100 RCs or more as the location of the first checkpoint, 'extremely short' simulation runs completely disappear, while 'extremely short' simulation runs still exist if a minimum number of 10 RCs as the location of the first checkpoint, suggested in [89], is used.

In Chapter 4, we also extended a stopping rule by adopting $n_1 = 100$ RCs as the location of the first checkpoint, based on experimentally obtained results, especially for a sequential RC method which helps to achieve final results with the required statistical error and the required level of coverage. The experimental coverage was significantly improved by adopting the proposed stopping rule for a sequential RC method, being comparable with NOBM and SA/HW, while the coverage of 'too short' simulation runs (including 'extremely short' runs) obtained without it is surprisingly poor, all below 21%, compared with the assumed theoretical confidence level of 95%.

The stopping rule for a sequential RC method and the rules for experimental coverage analysis for sequential simulation proposed in Chapter 2 have been applied to the analysis of the steady-state means for the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems. These results clearly show that a remarkable improvement in the quality of the sequential method of RCs was obtainable in the sense of the final coverage. In such a case, the restriction of the minimum number of RCs as the location of the first checkpoint played a crucial role. Our experimental results indicated that the sequential method of RCs is an attractive solution for simulation practitioners if special cares are taken to avoid 'too short' simulation runs, to choose an appropriate location of the first checkpoint, and to identify the regenerative state.

In Chapter 5, we considered sequential quantile estimations in steady-state simulations. Quantiles are often used to give a more complete description of the distribution, since one statistic - for instance, the mean value of a random variable - is seldom sufficient as a summary of an entire distribution. However, the direct estimation of quantiles, based on storing and consecutive multiple sorting of entire sequences of observations collected during the sequential steady-state simulation appears to be impractical in real applications of a stochastic simulation. A few approaches for estimating quantiles, which only require linear computation time and little memory, have been proposed so far.

In particular, we studied the properties of three sequential quantile estimators: *linear* and *batching* QE for the method of RCs, and *spectral* P^2 QE for the method of non-RCs, to determine the best one to implement in a fully automated simulation package such as Akaroa-2 [28]. As our results show, only the *batching* QE approach offers a reasonable quality of the final results, in terms of coverage analysis and bias, when estimating the response times in an $M/M/1/\infty$ queueing system. However, this does require easy recognition of the RCs in a simulated system. Otherwise, the *spectral* P^2 QE approach, which produces the poor coverage caused by slightly greater quantiles, can be an alternative since the poor coverage can be improved by assuming a higher statistical error of the final results.

We also studied two statistical tests to determine the suitability of applying them in the case of QE: Schruben's test [162] and the GSS test [57], which were originally developed for detecting the initial transient period when estimating the steady-state mean. Our results show that these tests also work when estimating the steady-state quantiles. Schruben's test appears to be better, since the empirical quantiles of the waiting times in the queue obtained with it are much closer to their theoretical steady-state quantiles with smaller errors.

One of the inherent problems in sequential steady-state simulations, which is that any simulation run can be very short since the stopping condition can be accidentally and temporarily satisfied, also happened when estimating quantiles. These short runs degrade the quality of the final result. This problem was solved by applying the rules of experimental coverage analysis (discussed in Chapter 2) for the three QE approaches. The other problem, especially in the sequential method of RCs, is that 'extremely short' simulation runs are produced by choosing an improper location of the first checkpoint. This problem also occurred when estimating quantiles using the *linear* QE and *batching* QE approaches based on the method of RCs. This problem was solved by choosing a proper location of the first checkpoint, after collecting at least 1,000 RCs.

The two recommended heuristic rules: Rules I and V, proposed for preventing 'too short' simulation runs being included in the final results (in Chapter 3), were applied for the three QE approaches. These rules work well for the *linear* QE and *batching* QE approaches, but they do not work for the *spectral* P^2 QE approach since this approach produces slightly greater quantiles.

Simulations of even moderately complex systems can require excessive computing time to obtain statistically accurate results. Advanced technologies of fast processors and large memories have helped those computationally intensive simulations. However, the stochastic nature of discrete-event simulation still creates a significant computational problem with increasing model complexity. Therefore, parallel and distributed simulation holds great promise for meeting the simulation needs of developers of increasingly complex systems, since the use of parallel and distributed computer systems significantly speeds up run times, enabling a simulation program to execute on a computing system containing multiple processors, such as personal computers interconnected by a communication network. Although parallel and distributed simulations have led to many important results in different domains, a robust and effective general methodology for dealing with various complex models has not yet been produced.

In Chapter 6, we studied parallel and distributed discrete-event simulations based on the MRIP scenario in terms of the speedup achievable when estimating mean values and quantiles using the methods of RCs and non-RCs. Empirical results obtained using Akaroa-2 showed quite good agreement with the *truncated Amdahl's law*, which was derived for estimating the theoretical speedup obtainable under the MRIP scenario in [133], for the speedup of steady-state simulations.

In general, the optimal speedup under the MRIP scenario when estimating mean values and quantiles can be achieved if the total run-length of the simulation is very long, the relative length of the initial transient period is very small or zero, and the distance between checkpoints is short. The speedup when using the method of RCs to estimate mean values and quantiles is not affected by the relative length of the initial transient period, but it can be limited by the location of the first checkpoint.

7.2 Suggestions for Future Research

In this thesis we have investigated some statistical issues that underlie the estimation of credible final results in sequential steady-state simulations. To obtain credible final results in fully automated sequential steady-state simulations, a host of problems remain unanswered. The following suggestions for future research are related to the original contributions of this thesis to find out better methods or techniques for a fully automated simulation package which can produce more credible simulation results.

- No simulation output data analysis methods can ensure that the final CIs from the sequential stochastic simulation will exactly contain the theoretical value with a probability equal to the assumed confidence level. One of the ongoing problems of research in this area is to find a valid method of output data analysis (in the sense of coverage) for simulations of highly dynamic stochastic processes, such as heavily loaded queueing systems and telecommunication networks.

- The *batching* QE approach based on the RC method offers good quality final results. However, this approach requires the stochastic process of interest to have (frequently occurring) regeneration points. This property is not shared by many real-world systems, such as a queueing system with two or more non-Poisson arrival streams. Therefore, the *batching* QE approach should be modified for the non-RC method.

- The *spectral* P^2 QE approach for the method of non-RCs produces the poor coverage caused by slightly greater quantiles. However, it can be applied for any system, since it does not require the recognition of the RCs in a simulated system. Therefore, the *spectral* P^2 QE approach should be further investigated.

- All heuristic rules and statistical tests for detecting the initial transient period have been developed for the case where the steady-state mean of the system is estimated, but none have been developed for estimating steady-state quantiles. Fortunately, we have shown that the Schruben's test could be applied for estimating quantiles. However, the ultimate solution would be to find a method of determining the length of the initial transient period in the sense of probability distribution, which can be applied for means, quantiles, proportions, and etc.

- One usually looks at only a few quantiles or combinations of quantiles to obtain information about the location, shape, and dispersion of the distribution. However, the empirical cumulative distribution function is a summary of all the quantiles, that can be used to estimate the entire cumulative distribution function. Therefore, the estimation of distributions rather than quantiles should be investigated.

- One of the inherent problems in sequential steady-state simulations is that any simulation run can be 'too short' since the stopping condition can be accidentally and temporarily satisfied. This accidental satisfaction of the stopping condition is usually caused by the sudden decrease of the relative statistical error within two consecutive checkpoints. As a means of avoiding 'too short' runs, one could consider adopting a method in which the changes in the relative statistical error can be smoothed, for example by fitting a simple least squares line, before the stopping condition is checked.
- The central idea of RCs is to exploit the fact that, when $\{X(t) : t \geq 0\}$ is a regenerative process, random observations collected between successive regeneration points are independent and identically distributed. The theoretically required run-length at the low level of autocorrelation should be much shorter than the one at the high level of autocorrelation. Therefore, it may be possible to prove theoretically that the method of RCs has been particularly prone to early stopping.
- The long-range dependence discovered in telecommunication networks has received a great attention in recent years. However, we are not aware of any methods of simulation output data analysis have been tested on long-range dependent processes. Therefore, the problem with output data analysis of long-range dependent processes is an open question.

REFERENCES

[1] Abramowitz, M., and Stegun, I. A. *Handbook of Mathematical Functions.* Government Printing Office. New York: Dover Publications, Inc., 1964.

[2] Adam, N. R. Achieving A Confidence Interval for Parameters Estimated by Simulation. *Management Science 29*, 7 (1983), pp. 856–866.

[3] Ahmed, H., Ronngren, R., and Ayani, R. Impact of Event Scheduling on Performance of Time Warp Parallel Simulations. In *Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences* (1994), pp. 455–462.

[4] Allen, A. O. *Probability, Statistics, and Queueing Theory with Computer Science Applications.* Academic Press, 1978.

[5] Avramidis, A. N., and Wilson, J. R. Correlation-Induction Techniques For Estimating Quantiles in Simulation Experiments. *Operations Research 46*, 4 (1998), pp. 574–591.

[6] Ayani, R., and Rajaei, H. Event Scheduling in Window Based Parallel Simulation Schemes. In *Proceedings of the Fourth IEEE Symposium on Parallel and Distributed Processing* (1992), pp. 56–60.

[7] Bagrodia, R. L. Perils and Pitfalls of Parallel Discrete Event Simulation. In *Proceedings of the 1996 Winter Simulation Conference* (Orlando, Florida, 1996), J. M. Charles, D. J. Banner, and J. J. Swain (eds.), pp. 136–143.

[8] Becker, M., Beylot, A.-L., Damm, G., and Thang, W.-Y. Automatic Run-Time Choice for Simulation Length in MIMESIS. *RAIRO Recherche Opèrationnelle/Operations Research 33*, 1 (1999), pp. 93–115.

[9] Bergmann, R. Qualitative Properties and Bounds for the Serial Covariances of Waiting Times in Single-Server Queues. *Operations Research 27*, 6 (1979), pp. 1168–1179.

[10] Blyth, C. R. Approximate Binomial Confidence Limits. *Theory and Methods, Journal of the American Statistical Association 81*, 395 (1986), pp. 843–855.

[11] Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. *Time Series Analysis: Forecasting and Control.* 3rd ed. Prentice-Hall, Inc, 1994.

[12] Bratley, P., Fox, B. L., and Schrage, L. E. *A Guide to Simulation.* Springer-Verlag, 1983.

[13] Brillinger, D. R. Estimation of the Mean of a Stationary Time Series by Sampling. *Journal Appl. Prob. 10* (1973), pp. 419–431.

[14] Brillinger, D. R. *Time Series: Data Analysis and Theory.* Holden-Day, San Francisco, 1981.

[15] Calvin, J. M., and Nakayama, M. K. Using Permutations in Regenerative Simulations to Reduce Variance. *ACM Transactions on Modeling and Computer Simulation 8*, 2 (1998), pp. 153–193.

[16] Cash, C. R., Dippold, D. G., Long, J. M., and Pollard, W. P. Evaluation of Tests for Initial-Condition Bias. In *Proceedings of the 1992 Winter Simulation Conference* (1992), J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson (eds.), pp. 577–585.

[17] Chambers, J. M., Cleveland, W. S., Kleiner, B., and Tukey, P. A. *Graphical Methods for Data Analysis.* Wadsworth & Brooks/Cole: Pacific Grove, California, 1983.

[18] Chen, E. J., and Kelton, W. D. Simulation-Based Estimation of Quantiles. In *Proceedings of the 1999 Winter Simulation Conference* (1999), P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans (eds.), pp. 428–434.

[19] Chien, C. Batch Size Selection for the Batch Means Method. In *Proceedings of the 1994 Winter Simulation Conference* (Lake Buena Vista, Florida, 1994), J. D. Tew, S. Manivannan, D. A. Sadowski and A. F. Seila (eds.), pp. 345–352.

[20] Cox, D. R., and Smith, W. L. *Queues.* Methuen, London, 1961.

[21] Crane, M. A., and Iglehart, D. L. Simulating Stable Stochastic Systems:III. Regenerative Processes and Discrete Event Simulations. *Operations Research 23*, 1 (1975), pp. 33–45.

[22] Crane, M. A., and Lemoine, A. J. *An Introduction to the Regenerative Method for Simulation Analysis in Lecture Notes in Control and Information Sciences.* Springer Verlag, 1977.

[23] Daley, D. J. The Serial Correlation Coefficients of Waiting Times in a Stationary Single Server Queue. *Journal of the Aust. Math. Soc. 8* (1968), pp. 683–699.

[24] Damerdji, H. On the Batch Means and Area Variance Estimators. In *Proceedings of the 1994 Winter Simulation Conference* (Lake Buena Vista, Florida, 1994), J. D. Tew, S. Manivannan, D. A. Sadowski and A. F. Seila (eds.), pp. 340–344.

[25] Deriche, M., Huang, N. K., and Tsai, W. T. Dynamic Load Balancing Distributed Heterogeneous systems Under Stationary and Bursty Traffics. In *Proceedings of the 32rd Midwest Symposium on Circuits and Systems* (1990), pp. 669–672.

[26] Ewing, G. C., McNickle, D., and Pawlikowski, K. Multiple Replications in Parallel: Distributed Generation of Data for Speeding up Quantitative Stochastic Simulation. In *Proceedings of 15th IMACS*

World Congress on Scientific Computation, Modelling and Applied Mathematics (Berlin, Germany, 1997), vol. 6, pp. 397–402.

[27] Ewing, G. C., Pawlikowski, K., and McNickle, D. Akaroa 2.5 User's Manual. Tech. Rep. TR-COSC 07/98, Department of Computer Science, University of Canterbury, Christchurch, New Zealand, 1998.

[28] Ewing, G. C., Pawlikowski, K., and McNickle, D. Akaroa-2: Exploiting Network Computing by Distributing Stochastic Simulation. In *13th European Simulation Multiconference* (Warsaw, Poland, 1999), pp. 175–181.

[29] Ewing, G. C., Pawlikowski, K., and McNickle, D. Akaroa 2.6.1 User's Manual. Tech. rep., Department of Computer Science, University of Canterbury, Christchurch, New Zealand, 2000.

[30] Fishman, G. S. *Concepts and Methods in Discrete Event Digital Simulation.* John Wiley, New York, 1973.

[31] Fishman, G. S. Achieving Specific Accuracy in Simulation Output Analysis. *Communications of the ACM 20* (1977), pp. 310–315.

[32] Fishman, G. S. Grouping Observations in Digital Simulation. *Management Science 24*, 5 (1978), pp. 510–521.

[33] Fishman, G. S. *Priciples of Discrete Event Simulation.* John Wiley, New York, 1978.

[34] Fishman, G. S., and Moore, L. R. An Exhaustive Analysis of Multiplicative Congruential Random Number Generators With Modulus $M = 2^{31} - 1$. *SIAM Journal of Sci. Stat. Comput. 7* (1986), pp. 24–45.

[35] Fishwick, P. A. Web-Based Simulation: Some Personal Observations. In *Proceedings of the 1996 Winter Simulation Conference* (1996), J. M. Charles, D. J. Banner, and J. J. Swain (eds.), pp. 772–779.

[36] Fisz, M. *Probability Theory and Mathematical Statistics.* John Wiley & Sons, Inc. New York. 3rd Ed., 1967.

[37] Fox, B. Estimation and Simulation. *Management Science 24* (1978), pp. 860–861.

[38] Fox, B. L., Goldsman, D., and Swain, J. Spaced Batch Means. *Operations Research Letters 10* (1991), pp. 255–263.

[39] Fujimoto, R. M. Performance Measurements of distributed Simulation Strategies. *Transactions of The Society for Computer Simulation 6*, 2 (1989), pp. 89–132.

[40] Fujimoto, R. M. Optimistic Approach to Parallel Discrete Event Simulation. *Transactions of The Society for Computer Simulation 7*, 2 (1990), pp. 153–191.

[41] Fujimoto, R. M. *Parallel and Distributed Simulation Systems.* John Wiley and Sons, Inc., 2000.

[42] Fujimoto, R. M. Parallel Discrete Event Simulation. *Communications of the ACM 33*, 10 (Oct. 1990), pp. 30–53.

[43] Gafarian, A. V., Ancker, C. J., and Morisaku, T. Evaluation of Commonly Used Rules for Detecting Steady State in Computer Simulation. *Naval Research Logistics Quarterly 78* (1978), pp. 511–529.

[44] Gaither, B. Empty Empiricism. *ACM Performance Evaluation Review 18*, 2 (1990), pp. 2–3.

[45] Gelenbe, E., and Kushwaha, R. Dynamic Load Balancing in Distributed Systems. In *MASCOTS'94: Modelling, Analysis and Simulation International Workshop* (1994), pp. 245–249.

[46] Gelenbe, E., and Kushwaha, R. Dynamic Load Balancing in Distributed Systems. In *MASCOTS'94: Modelling, Analysis and Simulation International Workshop* (1994), pp. 245–249.

[47] Glynn, P. W. Coverage Error for Confidence Intervals Arising in Simulation Output Analysis. In *Proceedings of the 1982 Winter Simulation Conference* (1982), Highland, Chao, and Madrigal (eds.), pp. 369–375.

[48] Glynn, P. W. Importance Sampling For Monte Carlo Estimation of Quantiles. In *Proceedings of the Second International Workshop on Mathematical Methods in Stochastic Simulation and Experimental Design* (1996), pp. 180–185.

[49] Glynn, P. W., and Iglehart, D. L. Simulation Output Analysis using Standardised Time Series. *Mathematics of Operations Research 15* (1990), pp. 1–16.

[50] Glynn, P. W., and Whitt, W. Estimating the Asymptotic variance with batch means. *Operations Research Letters 10* (1991), pp. 431–435.

[51] Glynn, P. W., and Whitt, W. The Asymptotic Validity of Sequential Stopping Rules for Stochastic Simulations. *The Annals of Applied Probability 2*, 1 (1992), pp. 180–198.

[52] Goldsman, D., and Kang, K. Cramer-von Mises Variance Estimators for Simulations. In *Proceedings of the 1991 Winter Simulation Conference* (Phoenix, Arizona, 1991), B. L. Nelson, W. D. Kelton, and G. M. Clark (eds.), pp. 916–920.

[53] Goldsman, D., Meketon, M., and Schruben, L. W. Properties of Standardised Time Series Weighted Area Variance Estimators. *Management Science 36* (1990), pp. 602–612.

[54] Goldsman, D., and Schmeiser, B. W. Computational Efficiency of Batching Methods. In *Proceedings of the 1997 Winter Simulation Conference* (1997), S. Andradottir, K. J. Healy, D. H. Withers, and B. L. Nelson (eds.), pp. 202–207.

[55] Goldsman, D., and Schruben, L. W. Asymptotic Properties of Some Confidence Interval Estimators for Simulation Output. *Management Science 30* (1984), pp. 1217–1225.

[56] Goldsman, D., and Schruben, L. W. New Confidence Interval Estimators using Standardised Time Series. *Management Science 36* (1990), pp. 393–397.

[57] GOLDSMAN, D., SCHRUBEN, L. W., AND SWAIN, J. J. Tests for Transient Means in Simulated Time Series. *Naval research Logistics Quarterly 41* (1994), pp. 171–187.

[58] GUNTHER, N. J. *The Practical Performance Analyst: Performance-By-Design Techniques For Distributed Systems.* McGraw-Hill, 1998.

[59] HALD, A. *Statistical Theory with Engineering Applications.* John Wiley and Sons, Inc., 1952.

[60] HEIDELBERGER, P. Statistical Analysis of Parallel Simulations. In *Proceedings of the 1986 Winter Simulation Conference* (1986), J. Wilson, J. Henriksen, and S. Roberts (eds.), pp. 290–295.

[61] HEIDELBERGER, P., AND LEWIS, P. A. W. Regression-Adjusted Estimates for Regenerative Simulations, with Graphics. *Communications of the ACM 24*, 4 (1981), pp. 260–273.

[62] HEIDELBERGER, P., AND LEWIS, P. A. W. Quantile Estimation in Dependent Sequences. *Operations Research 32*, 1 (1984), pp. 185–209.

[63] HEIDELBERGER, P., AND WELCH, P. D. A Spectral Method for Confidence Interval Generation and Run Length Control in Simulations. *Communications of the ACM 25* (1981), pp. 233–245.

[64] HEIDELBERGER, P., AND WELCH, P. D. Adaptive Spectral Methods for Simulation Output Analysis. *IBM J. Res. Dev. 25* (1981), pp. 860–876.

[65] HEIDELBERGER, P., AND WELCH, P. D. Simulation Run Length Control in the Presence of an Initial Transient. *Operations Research 31* (1983), pp. 1109–1144.

[66] HESTERBERG, T. C., AND NELSON, B. L. Control Variates for Probability and Quantile Estimation. *Management Science 44*, 9 (1998), pp. 1295–1312.

[67] HOROWITZ, E., AND SAHNI, S. *Fundamentals of Data Structures in PASCAL.* Computer Science Press, 1984.

[68] Howard, R. B., Gallagher, M. A., Bauer, K. W., and Maybeck, P. S. Confidence Intervals for Univariate Discrete-Event Simulation Output Using the Kalman Filter. In *Proceedings of the 1992 Winter Simulation Conference* (Phoenix, Arizona, 1992), J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson (eds.), pp. 586–593.

[69] Hsu, J. C., and Nelson, B. L. Control Variates for Quantile Estimation. In *Proceedings of the 1987 Winter Simulation Conference* (1987), A. Thesen, H. Grant, and W. D. Kelton (eds.), pp. 434–444.

[70] Hsu, J. C., and Nelson, B. L. Control Variates for Quantile Estimation. *Management Science 36*, 7 (1990), pp. 835–851.

[71] Iglehart, D. L. Simulating Stable Stochastic System, V: Comparison of Ratio Estimators. *Naval Research Logistics Quarterly 22* (1975), pp. 553–565.

[72] Iglehart, D. L. Simulating Stable Stochastic Systems, VI: Quantile Estimation. *Journal of the Association for Computing Machinery 23*, 2 (1976), pp. 347–360.

[73] Jain, R. *The Art of Computer Systems Performance Analysis.* John Wiley & Sons, Inc., 1991.

[74] Jain, R., and Chlamtac, I. The P^2 Algorithm for Dynamic Calculation of Quantiles and Histograms Without Storing Observations. *Communications of the ACM 28*, 10 (1985), pp. 1076–1085.

[75] Jefferson, D., and Reiher, P. Supercritical Speedup. In *Proceedings of the 24th Annual Simulation Symposium* (1991), pp. 159–168.

[76] Jenkins, G. M., and Watts, D. G. *Spectral Analysis and Its Applications.* Holden-Day, San Francisco, 1968.

[77] Judge, J., Beadle, H. W. P., and Chicharo, J. Correlation of HTTP Response Packet Size and Estimating Confidence intervals for for Mean Packet Size and WWW Traffic Volume. In *Proceedings of APCC 1997* (Sydney, Australia, 1997), pp. 382–386.

[78] Kang, K., and Goldsman, D. The Correlation Between Mean and Variance Estimators in Computer Simulation. *IIE Transactions 22*, 1 (1990), pp. 15–23.

[79] Kelton, W. D., and Law, A. M. An Analytical Evaluation of Alternative Strategies in Steady-State Simulation. *Operations Research 32*, 1 (1984), pp. 169–184.

[80] Kelton, W. D., Sadowski, R. P., and Sadowski, D. A. *Simulation With Arena.* WCB/McGraw-Hill., 1998.

[81] Kennedy, W., Lÿnders, R., Pawlikowski, K., and Stacey, C. *Methodology of Stochastic Simulation for Studying Non-Stationary Behaviour of Telecommunication Networks.* Final Report to Telecom Corporation of NZ, University of Canterbury, Christchurch, New Zealand, 1994.

[82] Kim, H. K., and Chung, S. M. Parallel Logic Simulation Using Time Warp on Shared-Memory Multiprocessors. In *Proceedings Eighth International Parallel Processing Symposium* (1994), pp. 942–948.

[83] Kleijnen, J. P. C. *The Role of Statistical Methodology in Simulation, In* Methodology in Systems Modelling and Simulation . B. P. Zeigler et al. (eds.), North-Holland, Amsterdam, 1979.

[84] Kleinrock, L. *Queueing Systems.* John Wiley & Sons, Inc., Vol. 1, 1975.

[85] Knuth, D. E. *Art of Computer Programming.* Addison-Wesley, Vol. 2, 1969.

[86] Konas, P., and Yew, P. C. Parallel Discrete Event Simulation on Shared-Memory Multiprocessors. In *Proceedings of the 24th Annual Simulation symposium* (1991), pp. 134–148.

[87] Kreutzer, W. *System Simulation Programming: Styles and Language.* Addison-Wesley, 1986.

[88] Kumar, D., and Harous, S. An Approach to Study Performance Properties of Distributed Simulation. In *Proceedings of the Second IEEE Symposium on Parallel and Distributed Processing* (1990), pp. 866–869.

[89] Lavenberg, S. S., and Sauer, C. H. Sequential Stopping Rules for the Regenerative Method of Simulation. *IBM Journal of Research Development* (1977), pp. 545–558.

[90] Law, A. M., and Carson, J. S. A Sequential Procedure for Determining the Length of a Steady-State Simulation. *Operations Research 27* (1979), pp. 1011–1025.

[91] Law, A. M., and Kelton, W. D. Confidence Intervals for Steady-State Simulations, II: A Survey of Sequential Procedures. *Management Science 28*, 5 (1982), pp. 550–562.

[92] Law, A. M., and Kelton, W. D. Confidence Intervals for Steady-State Simulations, I: A Survey of Fixed Sample Size Procedures. *Operations Research 32* (1984), pp. 1221–1239.

[93] Law, A. M., and Kelton, W. D. *Simulation Modeling and Analysis.* McGraw-Hill, Inc., 2nd Ed., 1991.

[94] Law, A. M., and McComas, M. G. Secrets of Successful Simulation Studies. In *Proceedings of 1991 Winter Simulation Conference* (1991), B. L. Nelson, W. D. Kelton, and G. M. Clark (eds.), pp. 21–27.

[95] L'Ecuyer, P. Random Numbers for Simulation. *Communications of the ACM 33* (1990), pp. 85–97.

[96] L'Ecuyer, P. Good Parameters and Implementations for Combined Multiple Recursive Random Number Generators. *Operations Research 47* (1999), pp. 159–164.

[97] L'Ecuyer, P., and Simard, R. On the Performance of Birthday Spacings Tests with Certain Families of Random Number Generators. In *Mathematics and Computers in Simulation* (1999), To appear.

[98] Lee, J. R. Coverage Analysis of Sequential Regenerative Simulation. Tech. Rep. TR-COSC 02/99, Department of Computer Science, University of Canterbury, Christchurch, New Zealand, 1999.

[99] Lee, J. R., McNickle, D., and Pawlikowski, K. A Survey of Confidence Interval Formulae for Coverage Analysis. Tech. Rep. TR-COSC 04/98, Department of Computer Science, University of Canterbury, Christchurch, New Zealand, 1998.

[100] Lee, J. R., McNickle, D., and Pawlikowski, K. Sequential Estimation of Quantiles. Tech. Rep. TR-COSC 05/98, Department of Computer Science, University of Canterbury, Christchurch, New Zealand, 1998.

[101] Lee, J. R., McNickle, D., and Pawlikowski, K. Confidence Interval Estimators for Coverage Analysis in Sequential Steady-State Simulation. In *Proceedings of the 22nd Australasian Computer Science Conference* (Auckland, New Zealand, 1999), pp. 87–98.

[102] Lee, J. R., McNickle, D., and Pawlikowski, K. Quality of Sequential Regenerative Simulation. In *13th European Simulation Multiconference* (Warsaw, Poland, 1999), pp. 161–167.

[103] Lee, J. R., McNickle, D., and Pawlikowski, K. Quantile Estimation in Sequential Steady-State Simulation. In *13th European Simulation Multiconference* (Warsaw, Poland, 1999), pp. 168–174.

[104] Lee, J. R., McNickle, D., and Pawlikowski, K. Initial Transient Period Detection For Steady-State Quantile Estimation. *2000 Summer Computer Simulation Conference* (Vancouver, British Columbia, 2000), pp. 169–174, or No. S213 in the SCSC 2000 CD–ROM.

[105] Lee, J. R., Pawlikowski, K., and McNickle, D. Do Not Trust Too Short Sequential Simulation. In *Proceedings of the 1999 Summer Computer Simulation Conference* (Chicago, Illinois, 1999), pp. 97–102.

[106] Lee, J. R., Pawlikowski, K., and McNickle, D. Experimental Coverage Analysis of Interval Estimators for Sequential Stochastic Simulation. In *First Western Pacific/Third Australia-Japan Workshop on Stochastic Models* (Christchurch, New Zealand, 1999), pp. 340–348.

[107] Lee, J. R., Pawlikowski, K., and McNickle, D. Sequential Steady-State Simulation: Rules of Thumb for Improving Accuracy of The Final Results. In *11th European Simulation Symposium, ESS'99* (Erlangen, Germany, 1999), pp. 618–622.

[108] Lee, J. R., Pawlikowski, K., and McNickle, D. Experimental Coverage Analysis of Interval Estimators for Sequential Stochastic Simulation. In *Special Issue of Mathematical and Computer Modelling* (Submitted, 2000).

[109] Lewis, P. A. W., and Orav, E. J. *Simulation Methodology for Statisticians, Operations Analysts, and Engineers, Volume I.* Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, California, 1989.

[110] Lewis, P. A. W., and Ressler, R. L. Average Regression-Adjusted Controlled Regenerative Estimates. In *Proceedings of the 1991 Winter Simulation Conference* (Phoenix, Arizona, 1991), B. L. Nelson, W. D. Kelton, and G. M. Clark (eds.), pp. 921–926.

[111] Lin, L.-M., and Xie, W. Load-skewing Task Assignment To Minimize Communication Conflicts On Network Of Workstations. *Parallel Computing 26* (2000), pp. 179–197.

[112] Lin, Y. B. Parallel Independent Replicated Simulation on a Network of Workstations. *SIMULATION* (Feb. 1995), pp. 102–110.

[113] Lin, Y. B., and Lazowska, E. D. A Time-Division Algorithm for Parallel Simulation. *ACM Transactions on Modeling and Computer Simulations 1*, 1 (Jan. 1991), pp. 73–83.

[114] Matsumoto, M., and Nishimura, T. Mersenne Twister: a 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. *ACM Transactions on Modeling and Computer Simulation 8*, 1 (1998), pp. 3–30.

[115] McNickle, D. C. Estimating the Average Delay of the First N Customers in an $M/Erlang/1$ Queue. *Asia-Pacific Journal of Operational Research 8*, 1 (1991), pp. 44–53.

[116] McNickle, D. C., Pawlikowski, K., and Ewing, G. Experimental Evaluation of Confidence Interval Procedures in Sequential Steady-State Simulation. In *Proceedings of the 1996 Winter Simulation Conference* (1996), J. M. Charles, D. J. Banner, and J. J. Swain (eds.), pp. 382–389.

[117] Meketon, M., and Heidelberger, P. A Renewal theoretic Approach to Bias Reduction in Regenerative Simulations. *Management Science 28*, 2 (Feb 1982), pp. 173–181.

[118] Meketon, M. S., and Schmeiser, B. Overlapping Batch Means: Something for Nothing? In *Proceedings of the 1984 Winter Simulation Conference* (1984), S. Sheppard, U. Pooch, and D. Pegden (eds.), pp. 227–230.

[119] Mood, A. M., Graybill, F. A., and Boes, D. C. *Introduction to the Theory of Statistics.* McGraw-Hill: New York, 1974.

[120] Moore, D. S., and McCabe, G. P. *Introduction to the Practice of Statistics.* W. H. Freeman and Company, New York, 2nd Ed., 1996.

[121] Mota, E., Wolisz, A., and Pawlikowski, K. Sequential Batch Means Techniques for Mean Value Analysis in Distributed Simulation. In *13th European Simulation Multiconference* (Warsaw, Poland, 1999), pp. 129–134.

[122] Mota, E., Wolisz, A., and Pawlikowski, K. Comparing Overlapping Batch Means and Standardized Time Series under Multiple Replications in Parallel. In *14th European Simulation Multiconference* (Ghent, Belgium, 2000).

[123] NICOL, D., AND FUJIMOTO, R. Parallel Simulation Today. *Annals of Operations Research 53* (1994), pp. 249–285.

[124] PAGE, E. H. Beyond Speedup: PADS, the HLA and Web-Based Simulation. In *Proceedings of the 13th Workshop on Parallel and Distributed Simulation, PADS'99* (Atlanta, Georgia, 1999), pp. 2–9.

[125] PAGE, E. H., BUSS, A., FISHWICK, P. A., HEALY, K., NANCE, R. E., AND PAUL, R. J. Web-Based Simulation: Revolution or Evolution ? In *ACM Transactions on Modeling and Computer Simulation* (Accepted, 1999).

[126] PARK, S. K., AND MILLER, K. W. Random Number Generators: Good Ones are Hard to Find. *Communications of the ACM 31* (1988), pp. 1192–1201.

[127] PARZEN, E. Mathematical Consideration in the Estimation of Spectra. *Technometrics 3* (1961), pp. 167–190.

[128] PAWLIKOWSKI, K. Steady-State Simulation of Queueing Processes: A Survey of Problems and Solutions. *ACM Computing Surveys 22*, 2 (1990), pp. 122–170.

[129] PAWLIKOWSKI, K. Simulation Studies of Telecommunications Networks and Their Credibility. In *13th European Simulation Multiconference* (Warsaw, Poland, 1999), pp. 349–355.

[130] PAWLIKOWSKI, K., AND DE VERE, L. Speeding up Sequential Simulation by Relative Variance Reduction. In *In Proc. 8th Australian Teletraffic Research Seminar, ATRS'93* (RMIT, Telstra, 1993), pp. 203–212.

[131] PAWLIKOWSKI, K., EWING, G., AND MCNICKLE, D. C. Performance Evaluation of Industrial Processes in Computer Network Environments. In *Proceedings of the 5th European Concurrent Engineering Conference, ECEC'98* (Erlangen, Germany, 1998), pp. 160–164.

[132] Pawlikowski, K., Jeong, H. J., and Lee, J. R. On Credibility of Simulation Studies of Telecommunication Networks. In *IEEE Communications Magazine* (Accepted, 2000).

[133] Pawlikowski, K., and McNickle, D. Speeding up Quantitative Stochastic Simulation. *In preparation.*

[134] Pawlikowski, K., McNickle, D. C., and Ewing, G. Coverage of Confidence Intervals in Sequential Steady-State Simulation. *Simulation Practice and Theory 6* (1998), pp. 255–267.

[135] Pawlikowski, K., and Yau, V. Methodology of Stochastic Simulation for Performance Evaluation of Data Communication Networks. Tech. Rep. TR-COSC 03/93, Department of Computer Science, University of Canterbury, Christchurch, New Zealand, 1993.

[136] Pawlikowski, K., Yau, V., and McNickle, D. C. Distributed and Stochastic Discrete-event Simulation in Parallel Time Streams. In *Proceedings of the 1994 Winter Simulation Conference* (Lake Buena Vista, Florida, 1994), J. D. Tew, S. Manivannan, D. A. Sadowski and A. F. Seila (eds.), pp. 723–730.

[137] Porras, J., and Ikonen, J. Approaches To The Analysis Of Distributed Simulation. In *11th European Simulation Symposium, ESS'99* (Erlangen, Germany, 1999), pp. 551–555.

[138] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. *Numerical Recipes : The Art of Scientific Computing.* Cambridge, 1986.

[139] Raatikainen, K. E. E. Simulation-Based Estimation of Variability in Queueing Systems. In *International Journal of Simulation: Practice and Theory* (To appear).

[140] Raatikainen, K. E. E. Run Length Control for Simultaneous Estimation of Several Percentiles in Dependent Sequences. *Methodology and Validation in SIMULATION series of SCS 19*, 1 (1987), pp. 54–59.

[141] Raatikainen, K. E. E. Simultaneous Estimation of Several Percentiles. *SIMULATION 49*, 4 (1987), pp. 159–164.

[142] Raatikainen, K. E. E. Sequential Procedure for Simultaneous Estimation of Several Percentiles. *Transactions of The Society for Computer Simulation 7*, 1 (1990), pp. 21–44.

[143] Raatikainen, K. E. E. Run Length Control Using Parallel Spectral Method. In *Proceedings of the 1992 Winter Simulation Conference* (1992), J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson (eds.), pp. 594–602.

[144] Raatikainen, K. E. E. A Sequential Procedure for Simultaneous Estimation of Several Means. *ACM Transactions on Modeling and Computer Simulation 3*, 2 (1993), pp. 108–133.

[145] Raatikainen, K. E. E. Simulation Estimation of Dynamic Properties in Queueing Systems. Tech. Rep. C-1993-35, Department of Computer Science, University of Helsinki, FIN-00014 University of Helsinki, Finland, 1993.

[146] Raatikainen, K. E. E. Controlling the Precision of Estimated Means in Interactive Simulation. *Annals of Operations Research 53* (1994), pp. 485–505.

[147] Raatikainen, K. E. E. Experimental Evaluation of A Parallel Spectral Method for Run Length Control in Steady-State Simulation. Tech. Rep. C-1994-55, Department of Computer Science, University of Helsinki, FIN-00014 University of Helsinki, Finland, 1994.

[148] Raatikainen, K. E. E. Simulation-Based Estimation of Proportions. *Management Science 41*, 7 (1995), pp. 1202–1223.

[149] Różański, R. The Asymptotic Consistency and Efficiency of Fixed-Size Sequential Confidence Sets. *Probability and Mathematical Statistics 18*, 1 (1998), pp. 19–31.

[150] RANDHAWA, S. U., AND BAXTER, L. K. A Study in the Application of Schriber's Truncation Rule to Simulation Output. *Transactions of The Society for Computer Simulation 9*, 3 (1992), pp. 175–192.

[151] REGO, V. J., AND SUNDERAM, V. S. Experiments in Concurrent Stochastic Simulation: The EcliPSe Paradigm. *Journal of Parallel and Distributed Computing 14* (1992), pp. 66–84.

[152] REZVAN, M. Improving VBR Voice Performance in Integrated Services Broadband Wireless Networks. Master's thesis, Dept. of Computer Science, University of Canterbury, Christchurch, New Zealand, 1991.

[153] RIGHTER, R., AND WALRAND, J. C. Distributed Simulation of Discrete Event Systems. *Proceedings of the IEEE 77*, 1 (Jan. 1989), pp. 99–113.

[154] ROBINSON, S. An Heuristic Technique for Selecting the Run-Length of Non-Terminating Steady-State Simulations. *Simulation 65*, 3 (1995), pp. 170–179.

[155] RUBIN, D. B., AND SCHENKER, N. Efficiently Simulating the Coverage Properties of Interval Estimates. *Applied Statistics 35*, 2 (1986), pp. 159–167.

[156] SAUER, C. H. Confidence Intervals for Queueing Simulations of Computer Systems. *ACM Performance Evaluation Review 8*, 1-2 (1979), pp. 46–55.

[157] SCHMEISER, B. Batch Size Effects in the Analysis of Simulation Output. *Operations Research 30* (1982), pp. 556–568.

[158] SCHRIBER, T. J., AND ANDREWS, R. W. A Conceptual Framework for Research in the Analysis of Simulation Output. *Communications of the ACM 24*, 4 (April, 1981), pp. 218–232.

[159] SCHRUBEN, L. W. A Coverage Function for Interval Estimators of Simulation Response. *Management Science 26* (1980), pp. 18–27.

[160] SCHRUBEN, L. W. Detecting Initialization Bias in Simulation Output. *Operations Research 30*, 3 (1982), pp. 569–590.

[161] SCHRUBEN, L. W. Confidence Interval Estimation Using Standardised Time Series. *Operations Research 31* (1983), pp. 1090–1108.

[162] SCHRUBEN, L. W., SINGH, H., AND TIERNEY, L. Optimal Tests for Initialization Bias in Simulation Output. *Operations Research 31* (1983), pp. 1167–1178.

[163] SEILA, A. F. A Batching Approach to Quantile Estimation in Regenerative Simulations. *Management Science 28*, 5 (1982), pp. 573–581.

[164] SEILA, A. F. Estimation of Percentiles in Discrete Event Simulation. *SIMULATION 39*, 6 (1982), pp. 193–200.

[165] SHEDLER, G. S. *Regenerative Stochastic Simulation.* Academic Press, Inc., 1993.

[166] SONG, W.-M. T. *Estimators of the Variance of the Sample Mean: Quadratic Forms, Optimal Batch Sizes, and Linear Combinations.* PhD thesis, Department of Statistics, School of Industrial Engineering, Purdue University, 1988.

[167] STACEY, C., PAWLIKOWSKI, K., AND MCNICKLE, D. C. Detection and Significance of the Initial Transient Period in Quantitative Steady-State Simulation. In *Proceedings of the 8th Australian Teletraffic Research Seminar, ATRS'93* (Melbourne, Australia, 1993), pp. 193–202.

[168] STEIGER, N. M., AND WILSON, J. R. Improved Batching For Confidence Interval Construction In Steady-State Simulation. In *Proceedings of the 1999 Winter Simulation Conference* (1999), P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans (eds.), pp. 442–451.

[169] TANNER, M. *Practical Queueing Analysis.* The IBM McGraw-Hill Series, 1995.

[170] TOKOL, G., GOLDSMAN, D., OCKERMAN, D. H., AND SCHRUBEN, L. W. Standardised Time Series L_p-Norm Variance Estimators for Simulations. *Management Science 44*, 2 (1998), pp. 234–245.

[171] Tseng, S., and Fogg, B. J. Credibility and Computing Technology. *Communications of the ACM 42*, 5 (May 1999), pp. 39–44.

[172] Wagner, D. B., and Lazowska, E. Parallel Simulation of Queueing Networks: Limitations and Potentials. *Performance Evaluation Review 17* (1989), pp. 146–155.

[173] Walpole, R. E., and Myers, R. H. *Probability and Statistics for Engineers and Scientists.* Macmillan Publishing Co., Inc., New York, 2nd Ed., 1978.

[174] Welch, P. On the Relationship between Batch Means and Overlapping Batch Means. In *Proceedings of the 1987 Winter Simulation Conference* (1987), A. Thesen, H. Grant, and W. D. Kelton (eds.), pp. 320–323.

[175] Westaway, F. W. *Scientific Method: Its Philosophical Basis and Its Modes of Application.* Blackie & Son Limited, 3rd Ed., 1924.

[176] Whitt, W. Planning Queueing Simulations. *Management Science 35*, 11 (1989), pp. 1341–1366.

[177] Whitt, W. Simulation Run Length Planning. In *Proceedings of the 1989 Winter Simulation Conference* (1989), E. A. MacNair, K. J. Musselman, and P. Heidelberger (eds.), pp. 106–112.

[178] Whitt, W. The Efficiency of One Long Run Versus Independent Replications in Steady-state Simulation. *Management Science 37*, 6 (1991), pp. 645–666.

[179] Wilson, E. B. *An Introduction To Scientific Research.* McGraw-Hill Book Company. Inc., 1952.

[180] Wilson, J. R., and Pritsker, A. A. B. A Survey of Research on the Simulation Startup Problem. *Simulation 31* (1978), pp. 55–58.

[181] Yau, V., and Pawlikowski, K. AKAROA: A Package for Automatic Generation and Process Control of Parallel Stochastic Simulation. In *Proceedings 16th Australian Computer science Conference* (1993), pp. 71–82.

[182] Yücesan, E. Randomisation Tests for Initialization Bias in Simulation Output. *Naval Research Logistics Quarterly 40* (1993), pp. 643–663.

APPENDICES

Appendix A

Automated Simulation Package: Akaroa-2

The simulation package, Akaroa-2[1], is used as a performance evaluation tool in this dissertation. Akaroa-2 is the latest version of a fully automated simulation tool designed for running parallel and distributed stochastic simulations under the Multiple Replications In Parallel (MRIP) scenario in a local area network (LAN) environment [27], [28], [181].

A.1 Architecture of Akaroa-2

The Akaroa-2 system has three main components: *akmaster*, *akslave*, and *akrun*, plus three auxiliary components: *akadd*, *akstat*, and *akgui*; more detailed discussion can be found in [27] and [28]. The relationships between the three main components of Akaroa-2 are shown in Figure A.1. Each bold-

[1] The first version of Akaroa was designed at the Department of Computer Science, University of Canterbury in Christchurch, New Zealand, by Associate Professor K. Pawlikowski (Computer Science) and Victor Yau (Computer Science) and Dr. D. McNickle (Management). The latest version (Akaroa-2) is a reimplementation by Dr. Greg Ewing (Computer Science). The Akaroa-2 package can be freely downloaded for the purpose of teaching and non-profit research activities at universities and research institutes from http://www.cosc.canterbury.ac.nz.

outlined box represents one Unix process, and the connecting lines represent Transmission Control Protocol and Internet Protocol (TCP/IP) stream connections.

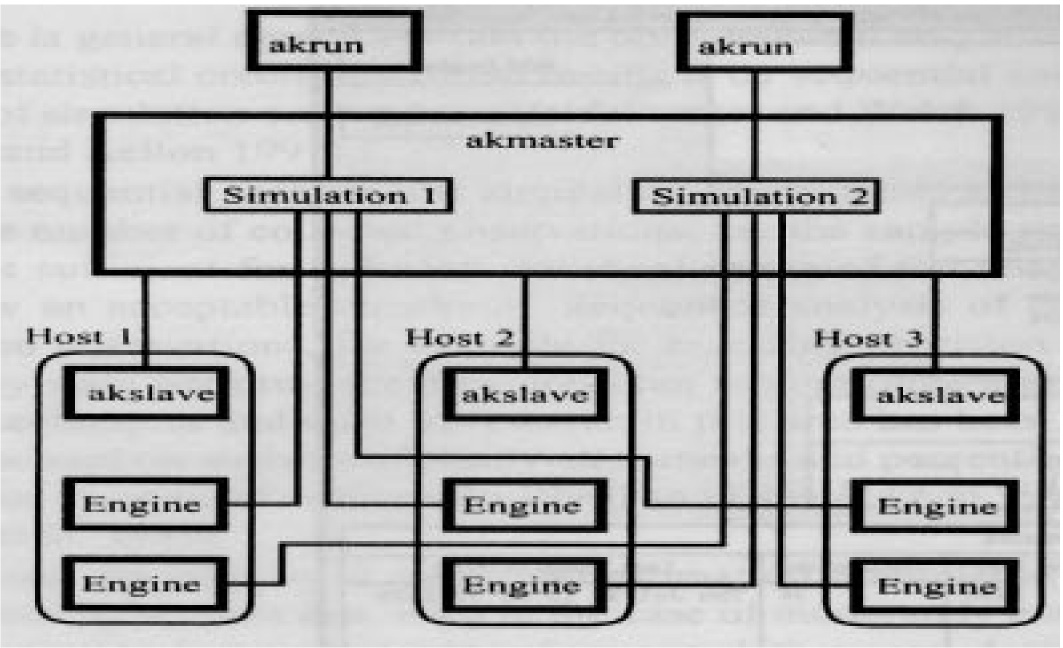


Figure A.1: Architecture of Akaroa-2 (taken from [28])

Akmaster is the master process which coordinates the activity of all other processes initiated by Akaroa-2. It launches new simulations, maintains state information about running simulations, performs global analysis of the data produced by simulation engines, and makes simulation stopping decisions.

The *akslave* processes run on hosts which run simulation engines. The sole function of the *akslave* is to launch simulation engine(s) on its host as directed by the *akmaster*.

Once the *akmaster* and any desired *akslaves* are running, the *akrun* program is used to initiate a simulation. It first contacts the *akmaster* process, obtaining its host name and port number from a file left by the *akmaster* in the user's home directory. For each simulation engine requested, the *akmaster* chooses a host from among those hosts on the LAN which are running *akslave* processes. It instructs the *akslave* on that host to launch an instance of the user's simulation program, passing on any specified arguments. The first time the simulation program calls one of the Akaroa-2 library routines, the simulation engine opens a connection to the *akmaster* process and identifies the simulation to which it belongs, so that the *akmaster* can associate the connection with the appropriate simulation data structure.

Akadd is used to add more simulations to a running simulation. This can be used to replace simulation engines which have been lost for some reason, or to speed up the simulation if more hosts become available. *Akstat* is used to obtain information about the state of the Akaroa-2 system: which hosts are available, which simulations are running, and what progress each simulation is making. *Akgui* provides a graphical user interface for starting and monitoring simulations that can be used instead of, or in addition to, *akrun* and *akstat*.

In the Akaroa-2 system, each engine performs sequential analysis of its own data to form a local estimate of each performance measure. At more or less regularly determined checkpoints, the engine sends its local estimates to the *akmaster* process, where the local estimates of each performance measure from all engines are combined to give a set of global estimates. Whenever a new global estimate is calculated, the relative statistical error is computed, and compared with the requested precision. When the precision of all analysed performance measures becomes satisfactory, the *akmaster* terminates all the simulation engines, and sends the final global estimates to the *akrun* process, which in turn reports them to the user.

A.2 Transient Period Detection in Akaroa-2

A number of ways to estimate the length of the initial transient period of steady-state simulations have been proposed; see Section 5.3.1 for more detailed discussion. Basic problems related to the existence of initial transient periods can be found, for example, in [128] and [150]. The length of the initial transient period has traditionally been determined using various heuristic rules.

More precise measures of the length of the initial transient period could be obtained by using various statistical tests to test the stationarity of data sequences. Each operates in a hypothesis testing framework, formally testing the null hypothesis that *there is no initialisation bias in the output mean* against the alternate hypothesis that initialisation bias in the output exists.

In Akaroa-2, a method applied for automatic detection of the length of the initial transient period was proposed by Pawlikowski [128]. It is based on the Schruben's test [162] using the SA/HW method for the variance estimator; see Section 5.3.1 for the Schruben's test and Appendix B.2 for the SA/HW method. This has been implemented in the simulation package Akaroa-2 [28].

In the case of steady-state simulation, a fully automated sequential statistical test for detecting the initial transient period in Akaroa-2 follows the following steps:

1. A rough, first approximation of the number of initial observations that should be discarded is obtained by applying a heuristic rule of thumb (labelled R5 in [128]).

    - the initial transient period is over after n observations $x_1, x_2, \cdots, x_n$ crosses the mean $\bar{X}(n)$ k times[2], where $\bar{X}(n) = \frac{1}{n}\sum_1^n x_i$.

2. Following the first rough selection of the transaction point for the initial data, the length of the initial transient period is more precisely determined sequentially by applying the statistical tests proposed by Schruben et al. in [162] for testing the stationarity of collected observations.

3. If the sequence of tested data cannot be considered stationary, it is discarded and the next sequence of observations tested. This process is repeated until the test determines that the system is free from the effect of the initial transient period, or some predefined upper limit on the simulation length is reached.

[2] This heuristic rule is sensitive to the value of k. The selection of $k = 25$ was adopted in Akaroa-2 as recommended in [43].

A.3 Random Number Generator in Akaroa-2

To achieve full credibility of simulation studies for the performance evaluation of a system one needs to use valid simulation models in valid simulation experiments. The most effective way of achieving this is to use good, thoroughly tested pseudo-random number generators (PRNGs).

It is a generally accepted and commonly used practice today to use algorithmic generators of (pseudo-random) uniformly distributed numbers as sources of basic randomness in a stochastic simulation. The most popular generators of simulation practice have belonged to a class of multiplicative linear congruential (MLC)-PRNGs, based on recursive algorithms in integer modulo M arithmetic. In today's world of 32-bit computers, MLC-PRNGs with a modulus of $M = 2^{31} - 1$ have focused special attention and, following exhaustive analysis, about 20 of them have been recommended as acceptable sources of independent and uniformly distributed pseudo-random numbers (see [34], [95], [96], [126]). These are the generators that have been used, for example, in GPSS (version H and PC), SIMSCRIPT II.5, SIMAN and SLAM II [93].

Akaroa-2 (version 2.4.1) used MLC-PRNGs[3] with a seed of $x_0 = 1$ and a modulus of $M = 2^{31} - 1$ whose 50 multipliers are taken from the top of the list of over 200 in [34]. The 50 multipliers used in Akaroa-2 (version 2.4.1) are listed in Section A.3.1. The *akmaster* process concatenates these 50 sequences into one sequence with a total length of about 10^{11} numbers; more detailed discussion can be found in [27] and [28]. This number has been used in our (computationally intensive) quality evaluation of the distributed estimators in Akaroa-2.

Recently, L'Ecuyer and Simard [97] have discovered that, when concerning the two-dimensional, $[0, 1)^2$, uniformity of random numbers generated by a LC-PRNGs, any LC-PRNGs fail the Birthday Spacing Test, if one applies this test to $n \geq 8\sqrt[3]{L}$ numbers generated by a given LC-PRNG, where L is the length of its cycle. This means that pseudo-random numbers should not be

[3] MLC-PRNG is given by $x_i = A * x_{i-1} \bmod M$, where A is the multiplier, M is the modulus, and x_0 is the seed.

used as a source of randomness in a single application if the simulation requires $n \geq 8\sqrt[3]{L}$ numbers. For example, a LC-PRNG with the cycle length of $L = 2^{31}$, when applying the rule of $n \geq 8\sqrt[3]{L}$, produces only 10321 acceptable pseudo-random numbers. This means that 516050 pseudo-random numbers, generated by MLC-PRNGs with 50 multipliers implemented in Akaroa-2 (version 2.4.1), can be used in a single simulation if pseudo-random numbers are used in pairs. However, in our applications we were concerned with one-dimensional uniformity in the interval $[0, 1)$. Therefore, the restriction imposed by [97] is not directly applicable in our studies. This allows us to claim that the sequence of 10^{11} pseudo-random numbers generated in Akaroa-2 (version 2.4.1) was sufficient for our research. The numbers of collected observations in a single simulation were always less than $3 * 10^6$.

However, using these MLC-PRNGs in real-life applications, for example, in simulation studies of networks fed by streams of teletraffic modelled by strongly autocorrelated processes, rare events simulations, and so on, can cause a potentially serious errors. These applications require very long samples of simulation output data to be collected or, equivalently, very long CPU time is needed for their generation to obtain final results with an acceptably small statistical error. Therefore, one obviously needs PRNGs of much longer cycles than those that would have been satisfactory two years ago.

Fortunately, PRNGs have been found that should be adequate in the foreseeable future for simulations demanding a long CPU time. A number of Multiple Recursive LC-PRNGs, and Combined Multiple Recursive LC-PRNGs, of cycles between 2^{185} to 2^{377}, can be found in [96], together with their portable implementations. Another discovery in the class of LC-PRNGs based recursions in polynomial arithmetic is known as the Generalised Feedback Shift Register PRNG (GFSFR-PRNG). A *twisted* GFSFR-PRNG, known as the Mersenne Twister, with a cycle $2^{19937} - 1$, and good virtual randomness in up to 623 dimensions, for up to 32-bit accuracy, has been proposed in [114], also with a portable implementation[4].

To cope with the recent requirements for a fully automated simulation tool,

[4] see http://www.math.keio.ac.jp/matumoto/emt.html

the PRNG in Akaroa-2 has recently[5] been changed to a Combined Multiple Recursive LC-PRNG described in [96]. The Combined Multiple Recursive LC-PRNG has two order 3 components as following (see [29] and [96]):

$$s1[n] = (a12 * s1[n-2] + a13 * s1[n-3]) \;\; mod \;\; m1, \tag{A.1}$$

and

$$s2[n] = (a21 * s2[n-1] + a23 * s2[n-3]) \;\; mod \;\; m2, \tag{A.2}$$

where $m1$ = 4294967087, $m2$ = 4294944443, $a12$ = 1403580, $a13$ = -810728, $a21$ = 527612, and $a23$ = -1370589. Then, using Equations (A.1) and (A.2), a pseudo-random number is obtained by

$$x[n] = \{((s1[n] - s2[n]) \;\; mod \;\; m1) + 1\}/(m1 + 1).$$

To see whether the new PRNG affects results presented in this dissertation, we have performed sequential coverage analysis using the method of SA/HW when estimating the mean response time in the $M/H_2/1/\infty$ queueing system only, by applying the principles of the sequential coverage analysis discussed in Chapter 2. We selected the $M/H_2/1/\infty$ queueing system to re-execute with the new PRNG, since this model theoretically requires many more observations to be collected than the $M/M/1/\infty$ and $M/D/1/\infty$ queueing systems. The theoretically required observations for those queueing systems can be found in Appendix F.

Each replication for coverage analysis was obtained with the required statistical error of 10% or less, and sequential coverage analysis was performed assuming that the required statistical error of the final result was 1% or less, both at a confidence level of 0.95. All numerical results obtained with the two different PRNGs: Multiplicative Linear Congruential (MLC)-PRNGs and Combined Multiple Recursive LC-PRNGs, are depicted in Figure A.2. As we can see in Figure A.2, the results of sequential coverage analysis obtained with the new PRNG is not significantly different from the results obtained with the MLC-PRNGs with a modulus of $M = 2^{31} - 1$.

[5] see [29]. The latest version of Akaroa-2 including the manual (version 2.6.1) updated on 8 August 2000 can be downloaded from http://www.cosc.canterbury.ac.nz.

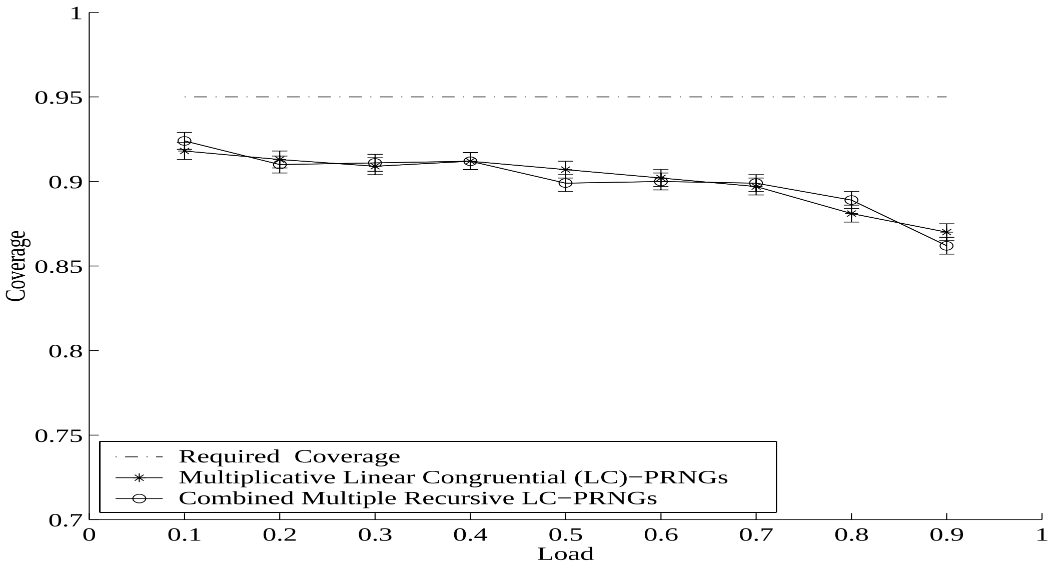


Figure A.2: Sequential coverage analysis using the method of SA/HW when estimating the mean response time in the $M/H_2/1/\infty$ queueing system with the two different PRNGs: Multiplicative Linear Congruential (MLC)-PRNGs and Combined Multiple Recursive LC-PRNGs (the confidence level = 0.95)

To evaluate the null hypothesis that the two PRNGs are equal, we have executed the statistical test of one way ANOVA (Analysis Of Variance) with the results shown in Figure A.2. The purpose of ANOVA is to assess whether the observed differences between the two PRNGs are statistically significant.

The calculations of the F statistic and its P value are organised in Table A.1, which contains numerical measures of the variation between PRNGs and within PRNGs. The *Model* and *Error* as sources of variation give information related to the variation between PRNGs and within PRNGs, respectively. The *Corrected Total* is the sum of the values for the *Model* and *Error*. Each *Sum of Squares* is a sum of squared deviations for the entries corresponding to the *Model*, *Error*, and *Corrected Total*. Each *Degrees of Freedom* is a degrees of freedom for the *Model* $(M-1)$, *Error* $(N-M)$, and *Corrected Total* $(N-1)$, where M is the number of groups and N is the number of observations in all groups. For each source of variation, the *Mean Square* is the sum of squares divided by the degrees of freedom. Each *Mean Square* for the *Model* and *Error*

Table A.1: Statistical test of one way ANOVA for the coverage obtained using the method of SA/HW when estimating the mean response time in the $M/H_2/1/\infty$ queueing system with the Multiplicative Linear Congruential (MLC)-PRNGs and Combined Multiple Recursive LC-PRNGs

Source	**Sum of Squares**	**Degrees of Freedom**	**Mean Square**	F **Value**	P **Value**
Model	0	1	MSM = 0	MSM/MSE = 0	0.967
Error	0.00454	16	MSE = 0.00028		
Corrected Total	0.00454	17	MST = 0.00026		

is called MSM and MSE, respectively. The MSM and MSE are estimates of the variance between PRNGs and within PRNGs. Then, to test the null hypothesis (H_0) in one way ANOVA, the F statistic is calculated by $F = MSM/MSE$. When H_0 is true, the F statistic has the F distribution with $(M-1, N-M)$ degrees of freedom, while when H_a is true, the F statistic tends to be large. We reject H_0 in favour of H_a if the F statistic is sufficiently large. The P value of the F test is the probability that a random variable having the F distribution with $(M-1, N-M)$ degrees of freedom is greater than or equal to the calculated value of the F statistic [120].

As shown in Table A.1, the result of the statistical test of one way ANOVA confirms that the two different PRNGs are not significantly different, since the P value of the F test is very large. Therefore, our results obtained with Multiplicative Linear Congruential (MLC)-PRNGs in this dissertation do not seem to be affected by their use in the reported research project.

A.3.1 The Multipliers used in Akaroa-2

The multipliers used by the MLC-PRNGs in Akaroa-2 (version 2.4.1) are listed below. They are taken from a list published by Fishman and Moore [34]. The ones marked * have been recommended by those authors as being of particularly high quality since they have satisfactorily passed a set of statistical tests [34].

No.	Multiplier	No.	Multiplier
1	742938285*	2	950706376*
3	1226874159*	4	6208991*
5	1343714438*	6	2049513912
7	781259587	8	482920380
9	1810831696	10	502005751
11	464822633	12	1980989888
13	329440414	14	1930251322
15	800218253	16	1575965843
17	1100494401	18	1647274979
19	62292588	20	1904505529
21	1032193948	22	1754050460
23	1580850638	24	1622264322
25	30010801	26	1187848453
27	531799225	28	1402531614
29	988799757	30	1067403910
31	1434972591	32	1542873971
33	621506530	34	473911476
35	2110382506	36	150663646
37	131698448	38	1114950053
39	1768050394	40	513482567
41	1626240045	42	2099489754
43	1262413818	44	334033198
45	404208769	46	257260339
47	1006097463	48	1393492757
49	1760624889	50	1442273554

Appendix B

Selected Methods of Sequential Simulation Output Data Analysis of Mean Values

Obtaining statistically valid final results by stochastic simulation is difficult because observations collected during the simulations are typically correlated, and the simulated process initially moves along a non-stationary trajectory.

Let us consider the sequence of observations $x_1, x_2, \cdots, x_n$ collected during a simulation run. The observations can be used to estimate the sample mean μ_x by calculating the arithmetic average of the sample:

$$\overline{X}(n) = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{B.1}$$

However, let us note that this estimate is a function of the sequence of random observations $x_1, x_2, \cdots, x_n$, and, as such, it assumes different, random values in different simulation experiments. Following a standard statistical approach, the accuracy of any such estimate can be assessed by considering the probability

$$Pr(\overline{X}(n) - \Delta_x(n) \leq \mu_x \leq \overline{X}(n) + \Delta_x(n)) = 1 - \alpha, \tag{B.2}$$

where $\Delta_x(n)$ is the half-width of the CI for the estimator, at an assumed confidence level $(1 - \alpha)$, $0 < \alpha < 1$.

On the basis of the *central limit theorem*[1], if observations $x_1, x_2, \cdots, x_n$ are realizations of independent and identically distributed random variables $X_1, X_2, \cdots, X_n$, one can have

$$\Delta_x(n) = t_{df,1-\alpha/2}, \hat{\sigma}[\overline{X}(n)], \tag{B.3}$$

where $t_{df,1-\alpha/2}$ is the $(1 - \alpha/2)$ quantile of the Student t-distribution with degrees of freedom $df = n - 1$, and $\hat{\sigma}^2[\overline{X}(n)]$ is the estimator of the variance of $\overline{X}(n)$, which is given by

$$\hat{\sigma}^2[\overline{X}(n)] = \frac{1}{n(n-1)} \sum_{i=1}^{n} (x_i - \overline{X}(n))^2. \tag{B.4}$$

Unfortunately, observations collected during simulations are usually not statistically independent. The general formula for the variance of the mean $\overline{X}(n)$ of observations $x_1, x_2, \cdots, x_n$ collected from a *covariance stationary*[2] process is

$$\hat{\sigma}^2[\overline{X}(n)] = \left[R(0) + 2 \sum_{k=1}^{n-1} (1 - \frac{k}{n}) R(k) \right] /n, \tag{B.5}$$

where

$$R(k) = E[(X_i - \mu_x)(X_{i-k} - \mu_x)], \quad 0 \leq k \leq n-1 \tag{B.6}$$

is the autocovariance of order k.

Neglecting the existing statistical autocorrelations, (that is equivalent to ignoring all the terms except $R(0)$ in Equation (B.5)), can lead to significant errors of estimation. For example, in an $M/M/1/\infty$ queueing system with 90% utilisation, the variance of the mean queue length calculated according to Equation (B.5) is 367 times greater than that from Equation (B.4); see [128]. Estimating $\sigma^2[\overline{X}(n)]$ without considering the autocorrelation among the observations would lead to either an excessively pessimistic or (more often),

[1] The *central limit theorem* states that as the sample size increases, the distribution of $\overline{X}(n)$ becomes closer to a normal distribution.

[2] A discrete-time stochastic process $X_1, X_2, \cdots$ is said to be *covariance stationary* if $\mu_i = \mu$ (for $i = 1, 2, \cdots$ and $-\infty < \mu < \infty$), $\sigma_i^2 = \sigma^2$ (for $i = 1, 2, \cdots$ and $\sigma^2 < \infty$), and $C_{i,i+j} = cov[X_i, X_{i+j}]$ is independent of i for $j = 1, 2, \cdots$ [93].

optimistic CI for μ_x. The estimation of the variance of the sample mean in autocorrelated processes is a major problem in assessing the CIs of the mean value during the stochastic simulation.

Various methods for data collection and analysis have been proposed to diminish the effect of the non-stationarity of simulated queueing processes (especially the initial non-stationarity caused by the initial transient period) and the autocorrelation of events (correlations among collected observations). These methods either try to weaken (or remove) autocorrelations among observations, or to exploit the correlated nature of observations in the estimation of variance needed for determining the CIs for the estimated parameters. Observations collected during the initial transient period neither belong to a stationary sequence nor characterise steady-state behaviour of the simulated process. Neglecting the existence of the initial transient period can also lead to significant bias in steady-state estimates of analysed performance measures. Various techniques for detecting the end point of the initial transient period can be found, for example, in [16], [57], and [128].

Many methods have been proposed to address the problems of autocorrelation and the initial transient period. Those relevant to this dissertation are:

- batch means
- methods based on spectral analysis
- regenerative cycles.

These three approaches are based on one ‘long’ replication, but differ from each other as they apply different approximations and data transformations for constructing CIs of the estimated parameters. Each method has its own merits and also potential difficulties. Hence, the quality of the final point and interval estimators produced may vary depending on the choice of output data analysis method.

B.1 Batch Means Methods

Various approaches based on the batch means (BM) have been proposed to discover the best options, such as the number of batches and batch sizes; see for example [38], [50], [90], [118], and [174]. Automated sequential simulation analysis procedures for implementing the BM can be found in [8], [128], and [168], and research on methods of BM under the MRIP scenario can be also found in [121].

The classical estimator known as Non-Overlapping Batch Means (NOBM) (we consider only NOBM which is also commonly called BM) is most widely used in simulation practice to calculate interval estimators from a single (long) simulation run by weakening correlations existing between consecutive data. NOBM requires that sequences of analysed data are stationary. Thus the initial transient observations, collected during the initial transient period, should be discarded. This approach is based on the assumption that observations more separated in time are less correlated. Thus, for sufficiently long batches of observations, the batch means are (almost) uncorrelated; see [13] for a formal justification.

The sequence of n original observations $x_1, x_2, \cdots, x_n$ is divided into non-overlapping batches $(x_{11}, x_{12}, \cdots, x_{1m})$, $(x_{21}, x_{22}, \cdots, x_{2m})$, $\cdots$ of each batch size m, sufficiently large so that the mean values over these batches are (almost) independent. Batch means $\overline{X}_1(m), \overline{X}_2(m), \cdots, \overline{X}_b(m)$, where

$$\overline{X}_i(m) = \frac{1}{m} \sum_{j=1}^{m} x_{ij}, \tag{B.7}$$

are used as (secondary) output data in the statistical analysis of the simulation results to obtain the mean and interval estimates of the process. The mean μ_x is estimated by

$$\overline{\overline{X}}(b, m) = \frac{1}{b} \sum_{i=1}^{b} \overline{X}_i(m), \tag{B.8}$$

where b is the number of batches.

A 100(1 - α)% CI for the steady-state mean μ_x obtained by applying the method of NOBM is given by

$$\overline{\overline{X}}(b,m) \pm t_{b-1,1-\alpha/2}\hat{\sigma}[\overline{\overline{X}}(b,m)], \tag{B.9}$$

where

$$\hat{\sigma}^2[\overline{\overline{X}}(b,m)] = \sum_{i=1}^{b} \frac{\{\overline{X}_i(m) - \overline{\overline{X}}(b,m)\}^2}{b(b-1)} \tag{B.10}$$

is the estimator of the variance of $\overline{\overline{X}}(b,m)$, and $t_{b-1,1-\alpha/2}$, for $0 < \alpha < 1$, is the upper $(1-\alpha/2)$ critical point from the Student t-distribution with degrees of freedom $b-1$.

The popularity of NOBM among practitioners continues because of the simplicity of the theory, regardless of reports of relatively poor coverage using this method, especially in heavily loaded systems. This is probably because sometimes batch sizes are accepted, even though they are not sufficiently large enough to obtain uncorrelated batch means. For example, one can select batches of as few as eight observations [33]. Song [166] showed a trade-off between bias and variance for a batch means estimator in accordance with batch sizes. Batches not having optimal numbers of observations can not guarantee that the final results are analysed properly within the NOBM method.

Determination of the optimal batch size and the number of batches are definitely problems for the batch means estimator. A few algorithms to determine the best number of batches b and the best batch size m, so that the batch means can be assumed to be independent and normally distributed, have been developed; see [19], [54], and [157].

Correlation between the batch means of the batch size m can be measured by estimators of the autocorrelation coefficients

$$\hat{r}(k,m) = \frac{\hat{R}(k,m)}{\hat{R}(0,m)}, \tag{B.11}$$

where

$$\hat{R}(k,m) = \frac{1}{b-k} \sum_{i=k+1}^{b} [\overline{X}_i(m) - \overline{X}(n)][\overline{X}_{i-k}(m) - \overline{X}(n)] \tag{B.12}$$

is the estimator of autocovariance of lag, $k = 0, 1, 2, \cdots$, in the sequence of batch means $\overline{X}_1(m), \overline{X}_2(m), \cdots, \overline{X}_b(m)$.

The sequential approach using the method of NOBM has been implemented in Akaroa-2 [28]. An algorithmic description in sequential simulation with this method, implemented in Akaroa-2, can be found in [128].

B.2 Methods Based on Spectral Analysis

Methods of variance estimation based on spectral analysis (SA) efficiently exploit the serial correlation between observations collected during one long simulation run. Analysed observations $x_1, x_2, \cdots, x_n$ must represent a stationary sequence, thus, as in NOBM, we assume that initial observations collected during the initial transient period have been discarded.

The autocovariance function $R(k)$ and the spectral density function $p_x(f)$ are closely related; more detailed discussion of their derivation can be found, for example, in [11], [14] and [76]. The spectral representation for the autocovariance function $R(k)$ can be shown as

$$R(k) = \int_{-1/2}^{1/2} p_x(f) \cos(2\pi f k) \ df. \tag{B.13}$$

The spectral density function $p_x(f)$ can be shown as

$$p_x(f) = \sum_{k=-\infty}^{\infty} R(k) \cos(2\pi f k), \quad -\infty \leq f \leq +\infty. \tag{B.14}$$

The variance $\sigma^2[\overline{X}(n)]$ can be obtained from Equation (B.5), which is given in terms of the autocovariance function $R(k)$. Assuming $\sum_{k=-\infty}^{\infty} |R(k)| < \infty$, we also have

$$\lim_{n\to\infty} n\sigma^2[\overline{X}(n)] = \sum_{k=-\infty}^{\infty} R(k) = p_x(0) \tag{B.15}$$

from Equations (B.5) and (B.14). Hence for sufficiently large n, the estimator of $\sigma^2[\overline{X}(n)]$ can be approximated from an estimator of the spectral density function $p_x(f)$ at frequency $f = 0$, i.e.

$$\sigma^2[\overline{X}(n)] \approx \frac{p_x(0)}{n}. \tag{B.16}$$

Several techniques have been proposed to obtain good estimators of the spectral density function $p_x(f)$. Most of them follow classical techniques of spectral estimation, based on the concept of spectral windows (special weighting functions introduced to lower the final bias of the estimators), for example, the Tukey-Hanning window; see [76] and the Parzen window; see [127]. However, the usefulness of spectral windows in reducing the bias of the estimate $\hat{p}_x(0)$ has been questioned in [63] and [64]. The spectrum is an even function, i.e., symmetric about zero. This means that it has either a peak or a valley at zero and is not approximately linear. Hence any weighted average of the spectrum about the point zero will result in a biased estimate of $p_x(0)$ and a larger region of averaging, i.e., the wider the spectral window, the more biased the estimate will be. Therefore, the spectral window should be narrow to lower the bias, but the variance of $p_x(0)$ increases as the width of the window decreases.

Another method based on spectral analysis to estimate the variance $\sigma^2[\overline{X}(n)]$ was developed by Heidelberger and Welch in [63] and [64]. This method estimates $p_x(0)$ from a regression fit to the logarithm of the average periodogram of the sequence of observations $x_1, x_2, \cdots, x_n$. The periodogram is a function of the discrete Fourier transform $A_x(j)$ of the observations, i.e.

$$I(\frac{j}{n}) = \frac{1}{n}|A_x(j)|^2 \tag{B.17}$$

and

$$A_x(j) = \sum_{s=1}^{n} x_s e^{-(2I\iota(s-1)j)/n}, \tag{B.18}$$

where $\iota = \sqrt{-1}$ and $0 < j < n/2$. The periodogram has the following approximate properties under very general conditions (see [63]);

$$E[I(\frac{j}{n})] \approx p_x(\frac{j}{n}), \quad 0 < j < n/2, \tag{B.19}$$

$$Var[I(\frac{j}{n})] \approx p_x^2(\frac{j}{n}), \quad 0 < j < n/2, \tag{B.20}$$

$$cov[I(\frac{j}{n}), I(\frac{i}{n})] \approx 0, \quad 0 < i \neq j < n/2. \tag{B.21}$$

A reasonable approach to obtain an estimate of $p_x(0)$ from the values of the periodogram in the region near zero is to assume the spectrum is a smooth function in this region and apply regression techniques. However, there are two problems associated with applying regression techniques to the periodogram: the variance is not constant and the exponential distribution is very positively skewed.

The former problem can be easily solved by taking the logarithm of the periodogram function. This has approximately the following properties (see [63]);

$$E[\log(I(\frac{j}{n}))] \approx \log(p_x(\frac{j}{n})) - 0.577, \quad 0 < j < n/2, \tag{B.22}$$

$$Var[\log(I(\frac{j}{n}))] \approx 1.645, \quad 0 < j < n/2, \tag{B.23}$$

$$cov[\log(I(\frac{j}{n})), \log(I(\frac{i}{n}))] \approx 0, \quad 0 < i \neq j < n/2. \tag{B.24}$$

The other problem of the positive skewness of the distribution in the periodogram can be reduced by averaging over adjacent periodogram values before taking the logarithm. The resulting function

$$L(f_j) = \log\{\left[I(\frac{2j-1}{n}) + I(\frac{2j}{n})\right]/2\} \tag{B.25}$$

for $f_j = (4j-1)/2n$ can be used in the application of regression techniques to estimate $p_x(0)$. Then, this function is approximated by a polynomial to obtain its value at zero. Finally, we can get the variance $\sigma^2[\overline{X}(n)]$ by applying the estimated value of $p_x(0)$ to Equation (B.16). (We will refer to this method as SA/HW after its authors.)

A $100(1-\alpha)\%$ CI for the steady-state mean obtained by applying the method of SA/HW is given by

$$\overline{X}(n) \pm t_{df,1-\alpha/2}\hat{\sigma}_{sp}[\overline{X}(n)], \tag{B.26}$$

assuming

$$\hat{\sigma}^2_{sp}[\overline{X}(n)] = \frac{1}{n}\hat{p}_x(0), \tag{B.27}$$

and $t_{df,1-\alpha/2}$, for $0 < \alpha < 1$, is the upper $(1-\alpha/2)$ critical point from the Student t-distribution with degrees of freedom df. There is no definitive method for choosing the parameter df, but the value of df depends here on the ratio of n/df_{max}, where df_{max} is the value of the upper lag considered in the autocovariance function $R(df)$; see [12] and [30].

The method of SA/HW [63] provides flexibility and stability in estimating $\sigma^2[\overline{X}(n)]$ and also produces quite accurate final results. The sequential approach using this method has been implemented in Akaroa-2 [28], and QNSim [146]. An algorithmic description in sequential simulation with this method, implemented in Akaroa-2, can be found in [128].

B.3 Regenerative Cycle Method

The method of regenerative cycles (RCs), first suggested by Cox and Smith [20], to analyse collected observations of the process $\{X(t) : t \geq 0\}$ has been systematically developed by a number of authors. The central idea of RCs is to exploit the fact that, when $\{X(t) : t \geq 0\}$ is a regenerative process, random variables between successive regeneration points are independent and identically distributed (i.i.d.). Thus it can circumvent the autocorrelation problem in estimates.

Let $\{X(t) : t \geq 0\}$ be a continuous time stochastic process. A definition of a regenerative process can be defined in terms of 'stopping times' for a stochastic process. A stopping time for a stochastic process $\{X(t) : t \geq 0\}$ is a random variable T taking values in $[0, +\infty)$. The random times $\{T_i : i \geq 0\}$ are said to be *regeneration points* (or *regenerative times*) for the process $\{X(t) : t \geq 0\}$, and $\{X(t) : T_{i-1} \leq t \leq T_i\}$ is said to be the i-th *cycle* of the process. The requirement that $\{T_i : i \geq 0\}$ be stopping times for $\{X(t) : t \geq 0\}$ means that for any fixed time t the occurrence of a regeneration point *prior* to time t (that is, $T_i \leq t$) may depend on the evolution of the process in the time interval

$(0, t]$ [165].

The RC method assumes that any regenerative process starts afresh (probabilistically) at each consecutive regeneration point. Thus, observations grouped into batches of random length, determined by successive regenerative points of the simulated process, are statistically independent, since the simulation always starts from a regenerative state, that is, the point at which its future state transitions do not depend on the past.

The method of RCs based on n RCs usually uses estimators in the form of a ratio of two variables. To estimate a steady-state mean μ_x of, for example, the waiting times in a queueing system, on the basis of observed waiting times $x_1, x_2, x_3, \ldots, x_n$ of consecutive customers, we are given the pairs of (secondary) output data $(a_1, y_1), (a_2, y_2), \ldots, (a_n, y_n)$. These are the realisations of i.i.d. random variables A_i and Y_i, $1 \leq i \leq n$, where A_i and Y_i denote, respectively, the number of customers processed and the sum of the waiting times in the ith RC. Let $\overline{y}(n)$, $\overline{a}(n)$, $s_{11}^2(n)$, $s_{22}^2(n)$, and $s_{12}^2(n)$ be the usual unbiased estimators for $E[Y]$, $E[A]$, $Var[Y]$, $Var[A]$, and $cov[Y, A]$ for any i, respectively; that is

$$\overline{y}(n) = \frac{1}{n}\sum_{i=1}^{n} y_i, \tag{B.28}$$

$$\overline{a}(n) = \frac{1}{n}\sum_{i=1}^{n} a_i, \tag{B.29}$$

$$s_{11}^2(n) = \frac{1}{n-1}\sum_{i=1}^{n} \left(y_i - \overline{y}(n)\right)^2, \tag{B.30}$$

$$s_{22}^2(n) = \frac{1}{n-1}\sum_{i=1}^{n} \left(a_i - \overline{a}(n)\right)^2, \tag{B.31}$$

and

$$s_{12}^2(n) = \frac{1}{n-1}\sum_{i=1}^{n} \left(y_i - \overline{y}(n)\right)\left(a_i - \overline{a}(n)\right). \tag{B.32}$$

As a consequence of the *strong law of large numbers*[3] for sequences of i.i.d. random variables, the point estimator of the mean

$$\hat{r}(n) = \frac{\overline{y}(n)}{\overline{a}(n)} \tag{B.33}$$

is a strongly consistent estimator of the steady-state mean μ_x, that is, $\hat{r}(n) \rightarrow \mu_x$ with probability one as $n \rightarrow \infty$. Moreover, the estimator for variance

$$s^2(n) = \{s_{11}^2(n) - 2\hat{r}(n)s_{12}^2(n) + \hat{r}^2(n)s_{22}^2(n)\} \tag{B.34}$$

is also strongly consistent, that is, $s^2(n) \rightarrow \sigma^2(n)$ with probability one as $n \rightarrow \infty$ [165].

A $100(1-\alpha)\%$ CI[4] for the steady-state mean μ_x obtained by applying the method of RCs based on n RCs is given by

$$\hat{r}(n) \pm \frac{s(n)t_{n-1,1-\alpha/2}}{\overline{a}(n)\sqrt{n}}, \tag{B.35}$$

where $t_{n-1,1-\alpha/2}$, for $0 < \alpha < 1$, is the upper $(1-\alpha/2)$ critical point from the Student t-distribution with degrees of freedom $n-1$.

As a consequence of the i.i.d. output data within consecutive RCs, the problems related with the initial transient period and the autocorrelations vanish simultaneously; more detailed discussion of RCs can be found in [21], [22], [71], and [165]. However, the random length of RCs makes the control of the accuracy of the final results more difficult. The various methods of RCs offer a very attractive solution to the main 'tactical' problems of stochastic simulation, but require a deeper *a priori* knowledge of the simulated processes. Usually a few, or even infinitely many, different sequences of regeneration points (for different types of regeneration states) can be distinguished in the behaviour of a system.

While the accuracy of the final simulation results from the method of RCs depends on the number of simulated RCs, the rate at which RCs occur depends

[3] *Strong law of large numbers for i.i.d. random variables:* Let $\{X_n : n \geq 1\}$ be a sequence of independent and identically distributed random variables, and set $S_n = X_1 + X_2 + \cdots + X_n$ for $n \geq 1$. If $E[|X|] < \infty$, then $S_n/n \rightarrow E[X]$ with probability one as $n \rightarrow \infty$ [165].

[4] The detailed derivation for constructing CIs can be found, for example, in [22] and [165].

on the simulated system. For example, in heavily loaded but stable queueing systems regenerative states can occur very rarely, making the method of RCs very ineffective, since it becomes difficult, if possible at all, to form a reliable point estimate and its CI. If a small number of RCs is recorded, the performance of this method appears to be poor indeed, worse than NOBM [91].

Our sequential implementation of the RCs method for the experimental studies in Akaroa-2 is based on the theory discussed here. A flowchart of the procedure is given in Figure B.1. The sequential algorithm is also described in the following section.

Sequential Procedure for the Method of RCs

The width of an estimated CI can be controlled by the use of an appropriate sequential stopping rule. Any sequential stopping rule, for example, based on a relative statistical error or an absolute statistical error, can be used in conjunction with the RC method. Among the possible criteria for stopping the experiment in the sequential RC method, we adopt a stopping criterion based on the relative half-width of the CIs at a given confidence level $(1-\alpha)$, defined as the ratio $\epsilon(n)$ in Equation (4.12) of Chapter 4. The simulation experiment is stopped when $\epsilon(n) \leq \epsilon_{max}$, where ϵ_{max} is the required limit of the relative statistical error of the results at the $100(1-\alpha)\%$ confidence level, $0 < \epsilon_{max} < 1$.

A sequential method of RCs is described below by the pseudocode using the following parameters [98]:

```
(1 - alpha) : The assumed confidence level of the final results
              (0 < alpha < 1)
Maximum Relative Statistical Error (epsilon_{max}) : The maximum
   acceptable value of the relative statistical error of the CIs
              (0 < epsilon_{max} < 1)


PROCEDURE RegenerativeAnalysis;
{Uses a ratio estimator for the method of regeneration cycles}
```

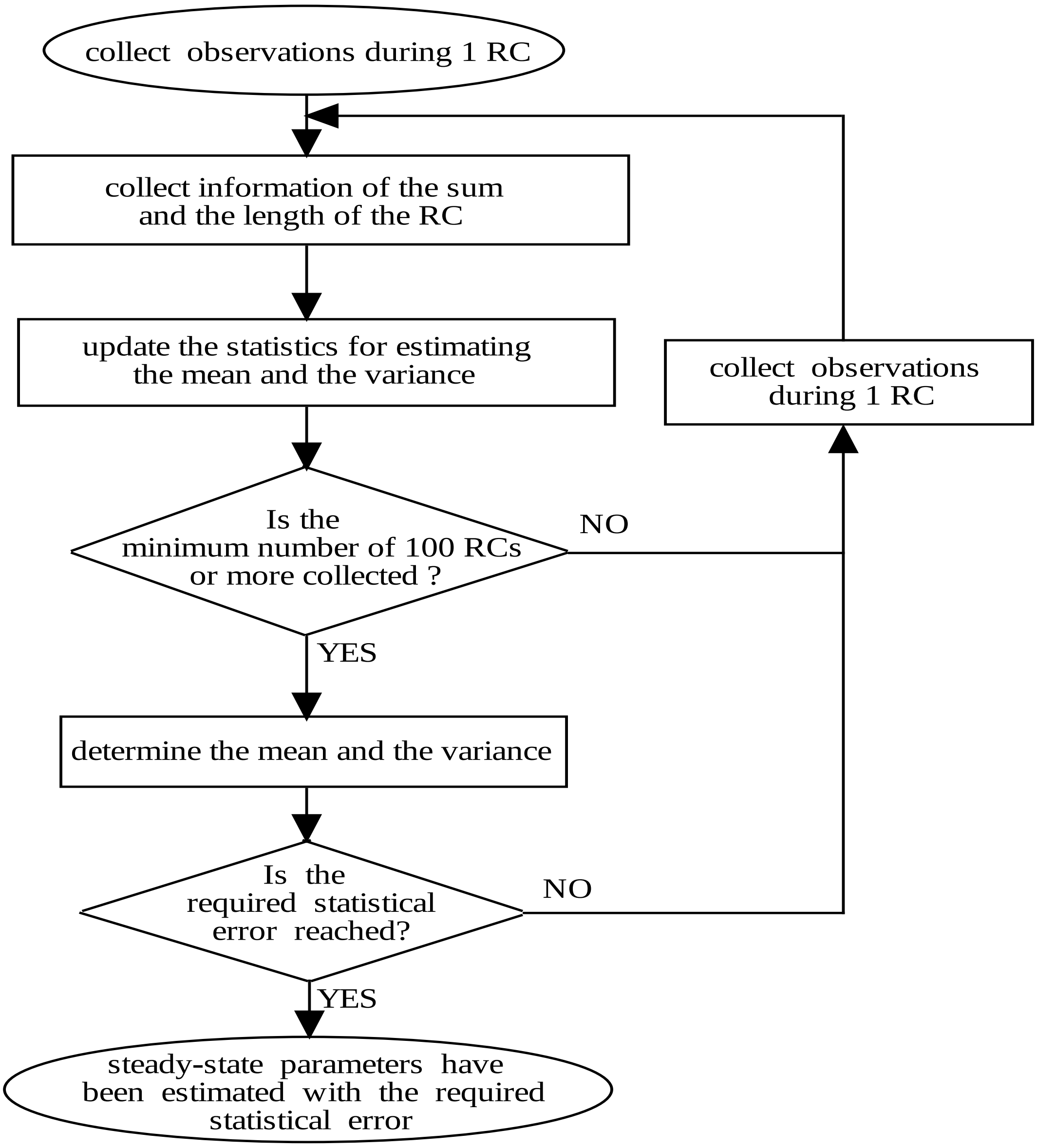

collect observations during 1 RC
collect information of the sum and the length of the RC
update the statistics for estimating the mean and the variance
collect observations during 1 RC
Is the minimum number of 100 RCs or more collected ?
NO
YES
determine the mean and the variance
Is the required statistical error reached?
NO
YES
steady-state parameters have been estimated with the required statistical error

Figure B.1: Flowchart for the sequential method of RCs

```
PROCEDURE GetNextRC;

  * Get an RC by collecting observations until a regeneration
    point is detected.

  * Collect information of the sum and the length of an RC.
```

```
     - RCSum
     - RCLength

  * Collect the following statistics for estimating the
       variance s^2(n) with RCSum and RCLength of RCs
     - MeanRCLength = SUM(RCLength) / NRCs;
     - MeanRCSums = SUM(RCSum) / NRCs;
     - SumofSqRCSums = SUM(RCSum*RCSum);
     - SumofSqRCLengths = SUM(RCLength*RCLength);
     - SumofRCSumbyRCLength = SUM(RCSum*RCLength);

END GetNextRC;

PROCEDURE UpdateStatistics;
{Update the overall variance and the mean using the
   classical estimator described in Chapter 4.
 The sums are updated dynamically, which is offering a
   quicker method for determining the overall variance.}

  * Update following statistics using formulae s^2_{11}(n),
      s^2_{22}(n), and s^2_{12}(n) described in the previous
      section.
    - VarTourSums = s^2_{11}(n);
    - VarTourLengths = s^2_{22}(n);
    - covariance = s^2_{12}(n);

  * Calculate the overall mean and overall variance using
      the classical estimator.
    - OverallMean = MeanRCSums / MeanRCLength;
    - OverallVariance = s^2(n);

END UpdateStatistics;
```

```
BEGIN {main procedure}

  {initialise parameters for calculating statistics from
       the collected observations in RCs}
  NRCs = 1;             {Number of RCs collected}
  RCSum = 0;            {Sum of the observations within an RC}
  RCLength = 0;         {Length of a single RC}
  MeanRCSums = 0.0;     {Overall mean of observations in RCs}
  MeanRCLength = 0.0;   {Overall mean of lengths of RCs}

  {For estimating the variance s^2(n)}
  SumofSqRCSums = 0.0;  {Sum of squares of sum of observations
                                                       in an RC}
  SumofSqRCLengths = 0.0;  {Sum of squares of length of an RC}
  SumofRCSumbyRCLength = 0.0; {Sum of length of an RC multiply
                                                by sum of an RC}

  {a condition of stopping the simulation has not been met yet}
  StopSimulation = false;

  while (not StopSimulation) {do}

    * Call GetNextRC;

    {The following procedures will be called after the minimum
     number of 100 RCs or more collected.}
    * Call UpdateStatistics;
    * Update the value of the relative statistical error using
        Equation (4.14) in Chapter 4.
      if (relative statistical error <=
                          Maximum Relative Statistical Error)
           StopSimulation = true;
      else StopSimulation = false;
  enddo;

END RegenerativeAnalysis;
```

Appendix C

Standardised Time Series Used in Statistical Tests for Detection of the Initial Transient Period

To estimate the variance of the sample mean of stationary observations we can use the central limit theorem to standardise i.i.d. random variables into an asymptotically standard normal random variable. Schruben originally introduced this idea in [161]. In this approach, a sequence of observations $x_1, x_2, \cdots, x_n$ is first divided into b contiguous batches of length m (assume $n = bm$); the observations $x_{(i-1)m+1}, x_{(i-1)m+2}, \cdots, x_{(i-1)m+m}$ comprise the ith batch, $i = 1, 2, \cdots, b$. Then, each batch is transformed into its standard form required by the functional central limit theorem, which is a generalisation of the central limit theorem.

We denote the grand mean by

$$\overline{X}(n) \equiv \frac{1}{n} \sum_{p=1}^{n} x_p. \tag{C.1}$$

For $i = 1, 2, \cdots, b$ and $j = 1, 2, \cdots, m$, the jth *cumulative mean* from batch i is

$$\overline{X}_i(j) \equiv \frac{1}{j} \sum_{p=1}^{j} x_{(i-1)j+p}. \tag{C.2}$$

(The quantity $\overline{X}_i(j)$ is called the ith *batch mean.*) For $i = 1, 2, \cdots, b$ and $0 \leq t \leq 1$, the *standardised time series* from batch i of length m is given by

$$T_{i,m}(t) \equiv \frac{\lfloor mt \rfloor (\overline{X}_i(m) - \overline{X}_i(\lfloor mt \rfloor))}{\sigma \sqrt{m}}, \tag{C.3}$$

where $\lfloor \cdot \rfloor$ is the greatest integer function and $\sigma^2 = \lim_{n \to \infty} n\sigma^2[\overline{X}(n)]$.

Schruben [161] shows that if observations $x_1, x_2, \cdots, x_n$ are a stationary sequence satisfying certain mild moments and ϕ-mixing conditions[1], then as $m \to \infty$ one can have

$$T_{i,m}(t) \to B(t), \quad 0 \leq t \leq 1, \tag{C.4}$$

a standard Brownian bridge process, which is a mathematical model of Brownian motion on the interval $[0, 1]$. All finite-dimensional joint distributions of B are normal and $\text{cov}(B(s), B(t)) = \min(s, t)(1\text{-}\max(s, t))$, $0 < s,\ t < 1$. Schruben also shows that $T_{i,m}(t)$ and $m\overline{X}_i(m)$ are asymptotically independent as the batch size m becomes large.

Schruben [161] proposed two estimators to estimate the variance of $\overline{X}(n)$ using two functions of $T_{i,m}(t)$: the maximum of $T_{i,m}(t)$, $0 \leq t \leq 1$, and the sum of $T_{i,m}(p/m)$, from $p = 1$ to m. Estimators using these two functions are known as the maximum estimator and the area estimator, respectively. These are as follows:

- the maximum estimator

$$\sigma^2_{max}[\overline{X}(n)] = \frac{1}{3b^2} \sum_{i=1}^{b} M_i, \tag{C.5}$$

  where $M_i = l_{i,max} \left[\overline{X}_i(m) - \overline{X}_i(l_{i,max})\right]^2 / (m - l_{i,max})$, and $l_{i,max} = \min\{l : T_{i,m}(l/m) \geq T_{i,m}(p/m),\ \text{ for } l = 1, 2, \cdots, m \text{ and } p = 1, 2, \cdots, m\}$, which is the location on [0,1] of the maximum of the i-th standardised time series, $1 \leq i \leq b$.

[1] The ϕ-mixing property (informally) means that, if the process runs for a sufficiently long time, observations in the distant past are approximately independent of those in the present [93].

An approximate $100(1-\alpha)\%$ CI for the steady-state mean μ_x obtained by applying the maximum estimator is given by

$$\overline{X}(n) \pm t_{3b,1-\alpha/2}\hat{\sigma}_{max}[\overline{X}(n)], \tag{C.6}$$

where $t_{3b,1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of a t-distribution with degrees of freedom $3b$; see [161].

- the area estimator

$$\sigma^2_{area}[\overline{X}(n)] = \frac{12}{(m^2-1)n^2}\sum_{i=1}^{b} A_i^2, \tag{C.7}$$

where $A_i = \sigma[\overline{X}_i(m)]\sqrt{m}\sum_{p=1}^{m} T_{i,m}(p/m)$.

An approximate $100(1-\alpha)\%$ CI for the steady-state mean μ_x obtained by applying the area estimator is given by

$$\overline{X}(n) \pm t_{b,1-\alpha/2}\hat{\sigma}_{area}[\overline{X}(n)], \tag{C.8}$$

where $t_{b,1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of a t-distribution with degrees of freedom b; see [161].

The standardised time series method is easy to apply and has some asymptotic advantage over the batch means method. However, selecting the batch size m is not easy and, while the property of ϕ-mixing is easy to assume, for many models it is difficult to prove. The major source of error for a standardised time series is in choosing too small a batch size m. Research on determining the best batch size for standardised time series, for both the simulation output data analysis and the initialisation bias test, continues both theoretically([49], [166]) and experimentally ([55], [57]).

The maximum estimator is asymptotically superior to the area estimator as $m \to \infty$; see [55]. There is also a claim that the standardised time series requires longer batches than the method of batch means [166]. The relationships between batch means and the area estimator, and comparisons of their efficiencies for large sample sizes can be found in [49]. A number of variants of the area estimator can be found in [53], [56], and [170]. There are also different approaches of combining the (weighted) area estimators or the maximum estimator with the batch means method [24] and [57].

Appendix D

Sequential Procedures for QE

This appendix presents the pseudocode of the sequential procedures for QE using the three approaches: *linear*, *batching*, and *spectral* P^2, described in Chapter 5. Among the possible criteria for stopping the experiment in the sequential QE method, we adopt a stopping criterion which is based on the relative half-width of the CIs at a given confidence level $(1 - \alpha)$, defined as the ratio $\epsilon(n)$ in Equation (5.4) of Chapter 5. The simulation experiment is stopped when $\epsilon(n) \leq \epsilon_{max}$, where ϵ_{max} is the required limit of the relative statistical error of the results at the $100(1-\alpha)\%$ confidence level, $0 < \epsilon_{max} < 1$.

D.1 Sequential QE Using the *Linear* Approach

Sequential procedures for QE using the *linear* approach are described below using the following parameters:

```
(1 - alpha) : The assumed confidence level of the final results
              (0 < alpha < 1)
Maximum Relative Statistical Error (epsilon_{max}) : The maximum
   acceptable value of the relative statistical error of the CIs
              (0 < epsilon_{max} < 1)
QuantileFactor : p of the p-quantile
GridSpacingFactor : The space between grid points
```

```
h+1 : The number of grid points (g(j), j = 0, 1, ..., h) and
      the grid points are spaced by the GridSpacingFactor
m : The number of RCs

PROCEDURE RegenerativeLinearQEAnalysis;

  PROCEDURE GetNextRC;

    * Get an RC by collecting observations until a regeneration
      point is detected.

    * Collect information of the number of observations in an RC
      that are less than or equal to grid points g(j) and the
      length of an RC.
      - NObsGrid_g(j);
      - TourLength;

    * Accumulate the sum and sum of squares of sample statistics.
      - NObsGrid_g(j) = NObsGrid_g(j) + NObsGrid_g(j-1);
      - SumofGrid_g(j) = SumofGrid_g(j) + NObsGrid_g(j);
      - SumofSqGrid_g(j) = SumofSqGrid_g(j) + (NObsGrid_g(j))^2;
      - SumofNumGrid_g(j) = SumofNumGrid_g(j) +
                            TourLength*NObsGrid_g(j);
      - SumofGridGrid_g(j) = SumofGridGrid_g(j) +
                             NObsGrid_g(j-1)*NObsGrid_g(j);
      - SumofNum = SumofNum + TourLength;
      - SumofSqNum = SumofSqNum + TourLength^2;

    * Increase the NRCs.
      - NRCs = NRCs + 1;

  END GetNextRC;
```

```
PROCEDURE UpdateStatistics;

  * Find the grid point on both sides of quantile estimate.
    - SumofGrid_g(j-1) <= SumofNum*QuantileFactor
                       <= SumofGrid_g(j);

  * Compute the sample cumulative distribution function at
    grid points g(j-1) and g(j).
    - SampleCDF_g(j-1) = SumofGrid_g(j-1) / SumofNum;
    - SampleCDF_g(j) = SumofGrid_g(j) / SumofNum;

  * Calculate the quantile estimate.
    - QuantileEstimate = g(j-1) + ((QuantileFactor -
                 SampleCDF_g(j-1))/(SampleCDF_g(j) -
                 SampleCDF_g(j-1)))*(g(j) - g(j-1));

  * Compute the mean and variance of the lengths of RCs.
    -  MeanRCs = SumofNum / NRCs;
    -  VarianceRCs = (NRCs*SumofSqNum - (SumofNum^2))
                        / (NRCs*(NRCs - 1));

  * Compute quantities that will be used to compute the
    variance of QuantileEstimate.
    - B_g(j) = ((NRCs*SumofSqGrid_g(j)) -
               (SumofGrid_g(j))^2)/(NRCs*(NRCs-1));
    - C_g(j) = ((NRCs*SumofNumGrid_g(j)) -
               (SumofNum*SumofGrid_g(j)))/(NRCs*(NRCs-1));
    - D_g(j) = B_g(j) - (2*SampleCDF_g(j)*C_g(j)) +
               (SampleVariance*SampleCDF_g(j)*SampleCDF_g(j));
    - V = (((QuantileEstimate - g(j))/(g(j-1) - g(j)))*D_g(j-1))
       + (((QuantileEstimate - g(j-1))/(g(j) - g(j-1)))*D_g(j));

  * Compute the density estimate.
```

```
      - DENS = (SampleCDF_g(j)-SampleCDF_g(j-1))/(g(j)-g(j-1));

    * Calculate the overall variance.
      - Variance = V/(MeanRCs^2 * DENS^2 * NRCs);

  END UpdateStatistics;

BEGIN {main procedure}

  {initialise parameters for calculating statistics from
       the collected observations in RCs}
  NRCs = 0;             {Number of RCs collected}
  TourLength = 0;       {Length of a single RC}
  NObsGrid_g(j) = 0;    {The number of observations in an RC
                           that are less than or equal to grid
                           points g_(j), j = 0, 1, ..., h}
  SumofGrid_g(j) = 0;   {Sum of NObsGrid_g(j) in an RC}
  SumofSqGrid_g(j) = 0; {Sum of squares of NObsGrid_g(j) in an RC}
  SumofNumGrid_g(j) = 0;{Sum of TourLength*NObsGrid_g(j) in an RC}
  SumofGridGrid_g(j) = 0; {Sum of NObsGrid_g(j-1)*NObsGrid_g(j)
                             in an RC}
  SampleCDF_g(j) = 0;     {The sample cumulative distribution
                             function at a grid point g(j)}
  SumofNum = 0;           {Sum of the length of an RC}
  SumofSqNum = 0;         {Sum of squares of the length of an RC}
  QuantileEstimate = 0.0; {The quantile estimate}
  Variance = 0.0;         {Variance of QuantileEstimate}
  MeanRCs = 0.0;          {Mean length of RCs}
  VarianceRCs = 0.0;      {Variance of MeanRCs}

  {a condition of stopping the simulation has not been met yet}
  StopSimulation = false;
  Call GetNextRC;

  while (not StopSimulation) {do}

    * Call GetNextRC;

    {Following procedures are called after the minimum number
      of m RCs collected.}
    * Call UpdateStatistics;
    * Update the value of the relative statistical error using
         Equation (5.4) of Chapter 5.
      if (relative statistical error <=
                            Maximum Relative Statistical Error)
           StopSimulation = true;
      else StopSimulation = false;

  enddo;

END RegenerativeLinearQEAnalysis;
```

D.2 Sequential QE Using the *Batching* Approach

Sequential procedures for QE using the *batching* approach are described below using the following parameters:

```
(1 - alpha) : The assumed confidence level of the final results
               (0 < alpha < 1)
Maximum Relative Statistical Error (epsilon_{max}) : The maximum
   acceptable value of the relative statistical error of the CIs
               (0 < epsilon_{max} < 1)
b : The batch size (i.e., the number of RCs in a batch)
r : The number of batches
```

D.2 Sequential QE Using the Batching Approach

```
PROCEDURE RegenerativeBatchingQEAnalysis;

  PROCEDURE GetNextBatch;

    * Get one batch of b RCs, producing observations x_(1),
      x_(2), ..., x_(m).

    * Compute three sample quantiles: SampleQuantile from all
      observations of x_(1), x_(2), ..., x_(m) in the i-th batch,
      SampleQuantile1 from the first half observations of x_(1),
      x_(2), ..., x_(m/2) in the i-th batch, and SampleQuantile2
      from the second half observations of x_(m/2+1), x_(m/2+2),
      ..., x_(m) in the i-th batch.

    * Calculate the jackknifed batch quantile.
      - Quantile = (2*SampleQuantile) -
                   ((SampleQuantile1 + SampleQuantile2)/2);

    * Accumulate the sum and sum of squares of Quantile.
      - SumofQuantile = SumofQuantile + Quantile;
      - SumofSqQuantile = SumofSqQuantile + Quantile^2;

    * Increase the BatchCount.
      - BatchCount = BatchCount + 1;

  END GetNextBatch;

  PROCEDURE UpdateStatistics;
  {Update the overall variance and the quantile.}

    * Calculate the overall quantile and overall variance.
      - QuantileEstimate = SumofQuantile/BatchCount;
      - Variance = (SumofSqQuantile-SumofQuantile^2/BatchCount)
```

```
                          / (BatchCount * (BatchCount-1));

  END UpdateStatistics;

BEGIN {main procedure}

  {initialise parameters for calculating statistics from
       the collected observations in a batch of b RCs}
  BatchCount = 0;         {Number of batches collected}
  Quantile = 0.0;         {The jackknifed batch quantile}
  SumofQuantile = 0.0;    {Sum of Quantile}
  SumofSqQuantile = 0.0;  {Sum of squares of Quantile}
  QuantileEstimate = 0.0; {The quantile estimate}
  Variance = 0.0;         {Variance of the estimator}

  {a condition of stopping the simulation has not been met yet}
  StopSimulation = false;

  Call GetNextBatch;

  while (not StopSimulation) {do}

    * Call GetNextBatch;

    {Following procedures are called after the minimum number
      of r batches collected.}
    * Call UpdateStatistics;
    * Update the value of the relative statistical error using
         Equation (5.4) of Chapter 5.
      if (relative statistical error <=
                           Maximum Relative Statistical Error)
           StopSimulation = true;
      else StopSimulation = false;
  enddo;

END RegenerativeBatchingQEAnalysis;
```

D.3 Sequential QE Using the *Spectral* P^2 Approach

Sequential procedures for QE using the *spectral* P^2 approach are described below using the following parameters:

```
(1 - alpha) : The assumed confidence level of the final results
               (0 < alpha < 1)
Maximum Relative Statistical Error (epsilon_{max}) : The maximum
   acceptable value of the relative statistical error of the CIs
               (0 < epsilon_{max} < 1)
QuantileFactor : p of the p-quantile

PROCEDURE SpectralP2QEAnalysis;

  PROCEDURE Initialization;

    * Set the increments in desired positions.
      - IncrementsPositions_(1) = 0;
      - IncrementsPositions_(2) = p/2;
      - IncrementsPositions_(3) = p;
      - IncrementsPositions_(4) = (1+p)/2;
      - IncrementsPositions_(5) = 1;

    * Set the desired positions.
      - DesiredPositions_(1) = 1;
      - DesiredPositions_(2) = 1+2p;
  - DesiredPositions_(3) = 1+4p;
  - DesiredPositions_(4) = 3+2p;
  - DesiredPositions_(5) = 5;

 * Set actual positions.
  - ActualPositions_(i) = i, for i = 1, ..., 5;

 * Set markers heights.
  - MarkerHeights_(i) = x_(i), for i = 1, ..., 5;

 * Initialize the QuantileEstimate and variance of
   QuantileEstimate.
  - QuantileEstimate = 0.0;
  - Variance = 0.0;

END Initialization;

PROCEDURE FindCell(x);

 * Find cell k such that (MarkerHeights_(k) <= x <=
  MarkerHeights_(k+1)) and adjust extreme values
  (MarkerHeights_(1) and MarkerHeights_(5)) if necessary.
  - case of x
    [x < MarkerHeights_(1)] MarkerHeights_(1)= x; k=1;
    [MarkerHeights_(1) <= x < MarkerHeights_(2)]  k=1;
    [MarkerHeights_(2) <= x < MarkerHeights_(3)]  k=2;
    [MarkerHeights_(3) <= x < MarkerHeights_(4)]  k=3;
    [MarkerHeights_(4) <= x < MarkerHeights_(5)]  k=4;
    [MarkerHeights_(5) < x] MarkerHeights_(5)= x; k=4;
   end case;

END FindCell;
```

```
PROCEDURE IncreaseActualPositions;

  * Increase actual positions of markers i = k+1 to 5.
    - ActualPositions_(i) = ActualPositions_(i) + 1;

END IncreaseActualPositions;

PROCEDURE IncreaseDesiredPositions;

  * Increase desired positions for all markers i = 1 to 5.
    - DesiredPositions_(i) = DesiredPositions_(i) +
                             IncrementsPositions_(i);

END IncreaseDesiredPositions;

PROCEDURE AdjustHeightsActualPositions;

  * Adjust heights and actual positions of markers i = 2 to 4.
    If (MarkerHeights_(i-1) < qt < MarkerHeights_(i+1)) is
    satisfied, MarkerHeights_(i) is calculated from the parabolic
    formula. Otherwise, MarkerHeights_(i) is calculated from the
    linear formula.
    - for i =2 to 4 do
       d = DesiredPositions_(i) - ActualPositions_(i);
                                  {offset of desired position}
       dp = ActualPositions_(i+1) - ActualPositions_(i);
                                  {offset of next position}
       dm = ActualPositions_(i-1) - ActualPositions_(i);
                                  {offset of previous position}
       qp = (MarkerHeights_(i+1) - MarkerHeights_(i)) / dp;
       qm = (MarkerHeights_(i-1) - MarkerHeights_(i)) / dm;

       if (d >= 1 and dp > 1) {
           qt = MarkerHeights_(i)+((1-dm)*qp+(dp-1)*qm)/(dp-dm);
           if (MarkerHeights_(i-1) < qt < MarkerHeights_(i+1))
               MarkerHeights_(i) = qt;
           else MarkerHeights_(i) = MarkerHeights_(i) + qp;
           ActualPositions_(i) = ActualPositions_(i) + 1;
        }
        else if (d <= -1 and dm < -1) {
           qt = MarkerHeights_(i)-((1+dp)*qm-(dm+1)*qp)/(dp-dm);
           if (MarkerHeights_(i-1) < qt < MarkerHeights_(i+1))
               MarkerHeights_(i) = qt;
           else MarkerHeights_(i) = MarkerHeights_(i) - qm;
           ActualPositions_(i) = ActualPositions_(i) - 1;
        }
       enddo;

END AdjustHeightsActualPositions;

PROCEDURE UpdateStatistics;

  * Return MarkerHeights_(3) as the estimate of p-quantile.
    - QuantileEstimate = MarkerHeights_(3);

  * Calculate density for variance of QuantileEstimate.
    - DENS = ((ActualPositions_(4)-ActualPositions_(2))/ObsCount)
             * ( ( (ActualPositions_(3) - ActualPositions_(2))/
                   (ActualPositions_(4) - ActualPositions_(3)) ) *
                   (MarkerHeights_(4) - MarkerHeights_(3)) +
                 ( (ActualPositions_(4) - ActualPositions_(3))/
                   (ActualPositions_(3) - ActualPositions_(2)) ) *
                   (MarkerHeights_(3) - MarkerHeights_(2)) );

  * Calculate variance of the estimator of mean using the spectral
    analysis method described in Appendix B.2.
```

```
      - sigma_sq = p_x(0)/ObsCount;

    * Calculate the variance of QuantileEstimate.
      - Variance = sigma_sq/(DENS^2);

  END UpdateStatistics;

BEGIN {main procedure}

  {Parameters for calculating the p-quantile using the Spectral
        P^2 approach}
  IncrementsPositions_(i);  {Increments of desired positions}
  DesiredPositions_(i);     {Desired positions}
  ActualPositions_(i);      {Actual positions}
  MarkerHeights_(i);        {Markers Heights}
  QuantileEstimate;         {Quantile Estimate}
  Variance;                 {Variance of QuantileEstimate}
  ObsCount;                 {Number of observations}

  {a condition of stopping the simulation has not been met yet}
  StopSimulation = false;

   * Discard n_0 observations collected from the initial transient
     period. The length of the initial transient period is
     determined by using Schruben test discussed in Section 5.3.1.

   * Collect the five observations (x_i, i = 1, ..., 5) and
     sort them in ascending order (x_(i), i = 1, ..., 5).
     - ObsCount = 5;

   * Call Initialization;

   * Decide the location of the first checkpoint.
```

```
     - w_1 = max[200, 2*n_0];

  while (not StopSimulation) {do}

    * Get an observation.
      - x = x_j, j = 6, ...;
      - ObsCount = ObsCount + 1;

    * Call FindCell(x);

    * Call IncreaseActualPositions;

    * Call IncreaseDesiredPositions;

    * Call AdjustHeightsActualPositions;

    {Following procedures are called when the checkpoint is reached.}
    * Call UpdateStatistics;
    * Update the value of the relative statistical error using
         Equation (5.4) of Chapter 5.
      if (relative statistical error <=
                            Maximum Relative Statistical Error)
           - StopSimulation = true;
      else { * Decide the next checkpoint.
               - w_1 = 3 * n_0;
               - StopSimulation = false;  }

  enddo;

END SpectralP2QEAnalysis;
```

Appendix E

Statistics from a Survey of Literature on Applications of Stochastic Simulation

More detailed results of a survey of technical literature over seven recent years, published in the Proceedings of INFOCOM, and over three recent years, published in three important technical journals: IEEE Transactions on Communications, IEEE/ACM Transactions on Networking, and Performance Evaluation, are depicted in Figures E.1 - E.4, and in Table E.1. The abbreviations used in these figures and tables are:

- TN: Total number of papers surveyed
- NS: Number of papers based on Simulation
- TS: Terminating Simulation
- SS: Steady-State Simulation
- US: Unspecified type of Simulation
- A1: Papers in which output data obtained from the TS are statistically analysed
- A2: Papers in which output data obtained from the TS are not properly analysed

- A3: Papers in which output data obtained from the SS are statistically analysed
- A4: Papers in which output data obtained from the SS are not properly analysed
- A5: Papers in which neither statistical analysis of the simulation output data nor the simulation type is mentioned.

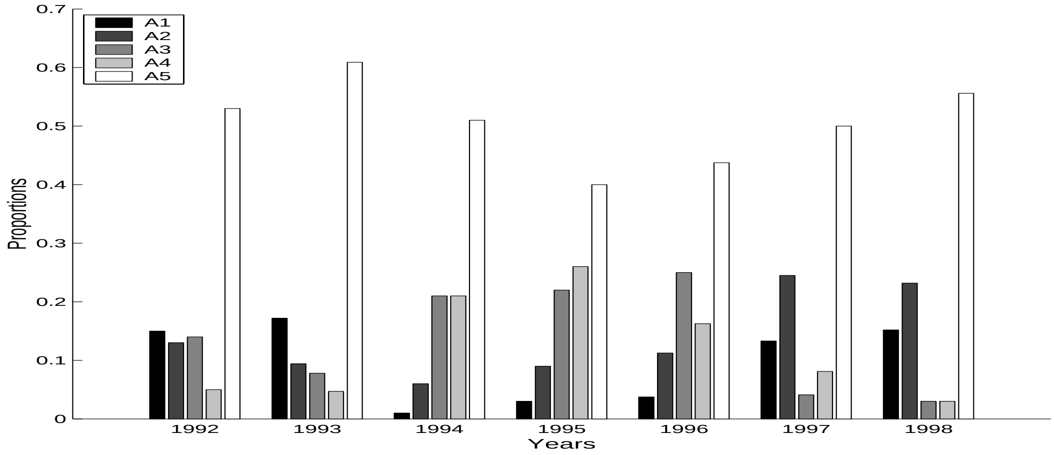


Figure E.1: Statistics of research papers, published in the Proceedings of IEEE INFOCOM, in which results were obtained by simulation

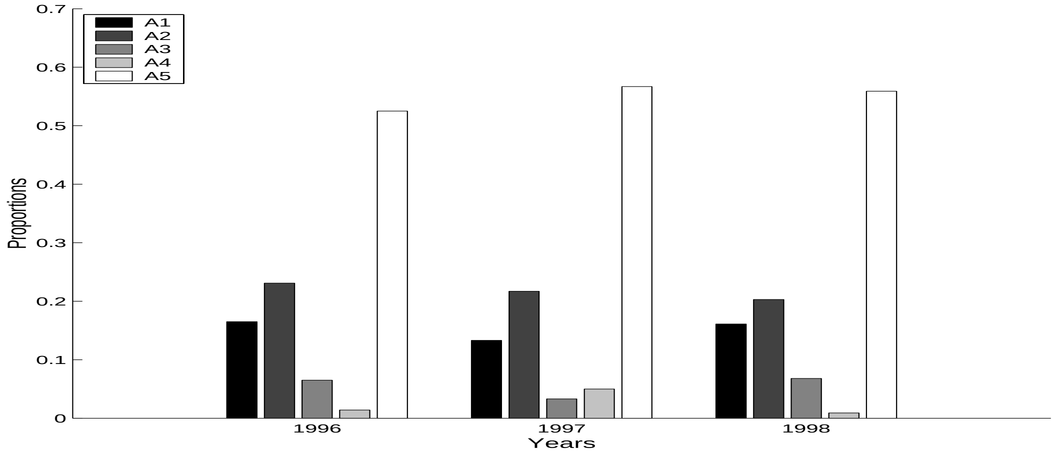


Figure E.2: Statistics of research papers, published in IEEE Transactions on Communications, in which results were obtained by simulation

The results presented in Figures E.1 - E.4, and in Table E.1 show that in about 56% of the surveyed papers, in which results were obtained by simulation, the authors do not even mention what statistical analysis method of the simulation output data or simulation type they used. Their final simulation results can not be acceptable as a scientific approach. Some other results can also be found in Chapter 1.

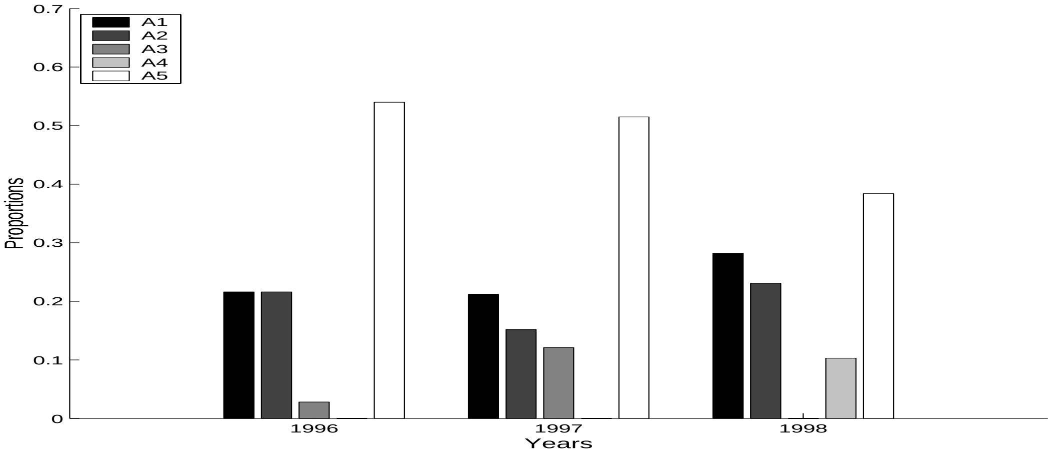


Figure E.3: Statistics of research papers, published in IEEE/ACM Transactions on Networking, in which results were obtained by simulation

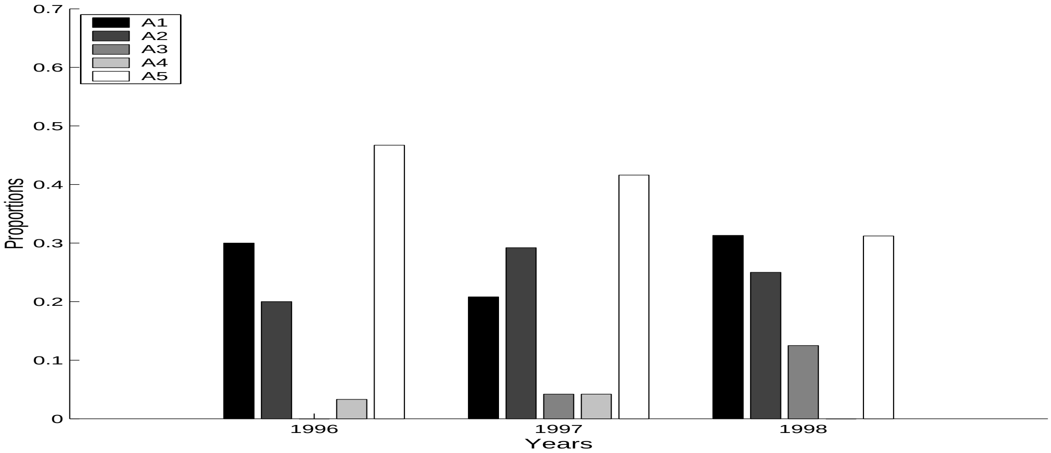


Figure E.4: Statistics of research papers, published in Performance Evaluation: An International Journal, in which results were obtained by simulation

Table E.1: Statistics of research papers published in the technical literature

(a) the Proceedings of IEEE INFOCOM

Year	*Total Number*	*Number of Papers*	*TS*		*SS*		*US*
	of Papers	*based on Simulation*	*A1*	*A2*	*A3*	*A4*	*A5*
1992	177	100	15	13	14	5	53
1993	167	64	11	6	5	3	39
1994	175	81	1	5	17	17	41
1995	156	77	2	7	17	20	31
1996	176	80	3	9	20	13	35
1997	168	98	13	24	4	8	49
1998	172	99	15	23	3	3	55

(b) IEEE Transactions on Communications

Year	*Total Number*	*Number of Papers*	*TS*		*SS*		*US*
	of Papers	*based on Simulation*	*A1*	*A2*	*A3*	*A4*	*A5*
1996	230	139	23	32	9	2	73
1997	227	120	16	26	4	6	68
1998	221	118	19	24	8	1	66

(c) IEEE/ACM Transactions on Networking

Year	*Total Number*	*Number of Papers*	*TS*		*SS*		*US*
	of Papers	*based on Simulation*	*A1*	*A2*	*A3*	*A4*	*A5*
1996	83	37	8	8	1	0	20
1997	80	33	7	5	4	0	17
1998	68	39	11	9	0	4	15

(d) Performance Evaluation Journal

Year	*Total Number*	*Number of Papers*	*TS*		*SS*		*US*
	of Papers	*based on Simulation*	*A1*	*A2*	*A3*	*A4*	*A5*
1996	66	30	9	6	0	1	14
1997	42	24	5	7	1	1	10
1998	37	16	5	4	2	0	5

Appendix F

Theoretically Required Simulation Run-Length for Some Stationary Queueing Systems

In a typical simulation, neither a variance nor a mean of parameters (such as waiting times, queue lengths, and so on) are known beforehand. Nevertheless, a simulation practitioner would like to plan a simulation and, in particular, estimate how long the simulation must be run so as to obtain a CI with the assumed statistical error. To help planning a simulation before any data has been collected, Whitt ([176]) has proposed that a required run-length is estimated from the approximation of the stochastic model of interest by a more elementary Markov model that can be analysed analytically. They have showed that some stochastic models can be approximated by *reflecting Brownian motion*[1] [176], [177].

However, the required simulation run-length for some stationary queueing systems can be calculated exactly. Depending on which steady-state parame-

[1] Reflecting Brownian motion is Brownian motion on the positive real line with constant negative drift, constant positive diffusion coefficient, and an impenetrable reflecting barrier at the origin.

ters of a queueing system are estimated in a sequential simulation, simulation run-lengths to satisfy the required confidence level with the acceptable statistical error are different. Here, we only consider two steady-state parameters: the mean waiting time in the queue and the mean response time from the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems, to calculate the theoretically required simulation run-length. The detailed derivation procedures are as follows.

F.1 Run-Length for the $M/M/1/\infty$ Queueing System

The derivation of a formula, which can calculate the theoretically required run-length for the $M/M/1/\infty$ queueing system, in this section follows the discussion in Daley [23].

First, we assume the sample mean waiting time in the queue from an $M/M/1/\infty$ queueing system is to be estimated with 5% relative statistical error for a 95% CI. Thus if $\hat{W}_q$ is the estimate of the sample mean waiting time in the queue, then we want

$$Pr(|\hat{W}_q - W_q| \leq 0.05W_q) = 0.95. \tag{F.1}$$

If $\rho = \lambda/\mu$ is the traffic intensity, then the theoretical mean steady-state waiting time in the queue can be obtained by

$$W_q = \frac{\rho^2}{\lambda(1-\rho)}, \tag{F.2}$$

and the theoretical variance of the waiting time in the queue is also obtained by

$$\sigma^2(W_q) = \frac{\rho^3(2-\rho)}{\lambda^2(1-\rho)^2}, \tag{F.3}$$

where λ is the arrival rate and μ is the service rate [73] (p. 525).

The Laplace-Stieltjes transform of the response times in the $M/G/1/\infty$

queueing system is defined as:

$$W^*(s) = \frac{(1-\rho)sB^*(s)}{s - \lambda[1 - B^*(s)]}, \tag{F.4}$$

where $B^*(s)$ is the Laplace-Stieltjes transform of a function $f(t)$, and also from the convolution property of transforms, it can be written as

$$W^*(s) = W_q^*(s)B^*(s), \tag{F.5}$$

where $W_q^*(s)$ is the Laplace-Stieltjes transform of the waiting time in the $M/G/1/\infty$ queueing system, since

$$ResponseTime = WaitingTimeintheQueue + ServiceTime. \tag{F.6}$$

Therefore, the Laplace-Stieltjes transform of the waiting times in the $M/G/1/\infty$ queueing system is

$$W_q^*(s) = \frac{s(1-\rho)}{s - \lambda[1 - B^*(s)]}; \tag{F.7}$$

see [84] for the detailed discussion.

From these results, the Laplace-Stieltjes transforms of the response time and waiting time for the $M/M/1/\infty$ queueing system can be easily obtained, since it is the special case of the $M/G/1/\infty$ queueing system with the squared coefficient of variation of the service time C_s^2 is one. The Laplace-Stieltjes transform $B^*(s)$ of the exponential service time for the $M/M/1/\infty$ queueing system is defined by

$$B^*(s) = \int_0^\infty e^{-st}\mu e^{-\mu t}dt = \frac{\mu}{\mu + s} \tag{F.8}$$

in [84] (p. 195). The Laplace-Stieltjes transform $W^*(s)$ of the response time for the $M/M/1/\infty$ queueing system is calculated from Equation (F.4) using Equation (F.8) as follows:

$$W^*(s) = \frac{\mu(1-\rho)}{s + \mu(1-\rho)}, \tag{F.9}$$

and the Laplace-Stieltjes transform $W_q^*(s)$ of the waiting time is also calculated from Equation (F.7) using Equation (F.8) as follows:

$$W_q^*(s) = \frac{(s+\mu)(1-\rho)}{s + (\mu - \rho)}, \tag{F.10}$$

in [84] (p. 202).

From Daley [23], if the system has been operating for a long time and one selects N observations with waiting times $W_1, \cdots, W_N$, then the sample mean waiting time in the queue

$$\hat{W}_q = \sum_{i=1}^{N} W_i/N \tag{F.11}$$

has, for sufficiently large N,

$$N\sigma^2(\hat{W}_q) \approx \sigma^2(W_q)\left[1 + 2\sum_{j=1}^{\infty} \rho_j(m)\right], \tag{F.12}$$

where

$$1 + 2\sum_{j=1}^{\infty} \rho_j(m) = \frac{1+\rho}{1-\rho} + \frac{\lambda(W_q^{'''} - W_q^{'}W_q^{''})}{(1-\rho)(W_q^{''} - W_q^{'}W_q^{'})} \tag{F.13}$$

(where $W_q^{'}$, $W_q^{''}$ and $W_q^{'''}$ can be obtained by the first, second and third differentiations[2] of the Laplace-Stieltjes transform $W_q^*(s)$, in Equation (F.10), of the waiting times in the queue, respectively, and $\sigma^2(W_q)$ can be calculated by $(W_q^{''} - W_q^{'}W_q^{'})$); see [23] and [32].

From Equation (F.1), we have

$$Pr\left(\frac{|\hat{W}_q - W_q|}{\sigma(\hat{W}_q)} \leq \frac{0.05W_q}{\sigma(\hat{W}_q)}\right) = 0.95 \tag{F.14}$$

or

$$\frac{0.05W_q}{\sigma(\hat{W}_q)} = 1.96. \tag{F.15}$$

From Equations (F.12) and (F.15), we can obtain the following equation

$$N = A(\rho)\left(\frac{1.96}{0.05W_q}\right)^2, \tag{F.16}$$

where

$$A(\rho) = \sigma^2(W_q)\left[\frac{1+\rho}{1-\rho} + \frac{\lambda(W_q^{'''} - W_q^{'}W_q^{''})}{(1-\rho)(W_q^{''} - W_q^{'}W_q^{'})}\right]$$

[2] This can be easily calculated using, for example, Maple.

$$= \frac{\rho^3(2-\rho)}{\lambda^2(1-\rho)^2}\left[\frac{2\mu^3+5\lambda\mu^2-4\mu\lambda^2+\lambda^3}{(2\mu-\lambda)(\mu-\lambda)^2}\right]; \qquad \text{(F.17)}$$

from the private communication with D. McNickle (2000), [23] and [32]. As simplifying Equation (F.16), the number of observations N required theoretically when estimating the mean waiting time in the $M/M/1/\infty$ queueing system with 5% of the relative statistical error for a 95% CI can be calculated by

$$N = 1536.64\left(\frac{2+5\rho-4\rho^2+\rho^3}{\rho(1-\rho)^2}\right). \qquad \text{(F.18)}$$

The numbers of observations required in theory, with a relative statistical error of 5% and 10% at a 95% CI, are presented in Table F.1.

Table F.1: Required run-length when estimating the mean waiting time in the $M/M/1/\infty$ queueing system at a 95% CI

ρ	*Relative Statistical Rrror = 5%*	*Relative Statistical Rrror = 10%*
0.1	46,687	11,671
0.2	34,190	8,547
0.3	33,105	8,276
0.4	36,357	9,134
0.5	44,562	11,140
0.6	60,441	15,110
0.7	94,710	24,830
0.8	189,775	47,443
0.9	681,072	170,268

F.2 Run-Length for the $M/D/1/\infty$ Queueing System

The required observations in theory for a sequential steady-state simulation when estimating the mean waiting time for the $M/D/1/\infty$ queueing system can be obtained from the $M/G/1/\infty$ queueing system, since the $M/D/1/\infty$ queueing system is a special case of the $M/G/1/\infty$ queueing system with the squared coefficient of variation of the service time C_s^2 is zero [169], [176]. For the $M/D/1/\infty$ queueing system, the theoretical mean waiting time in the queue can be obtained by

$$W_q = \frac{\rho E[s]}{2(1-\rho)}, \tag{F.19}$$

and the theoretical variance of the waiting time in the queue is also obtained by

$$\sigma^2(W_q) = \frac{\rho (E[s])^2}{3(1-\rho)} + \frac{\rho^2 (E[s])^2}{4(1-\rho)^2}, \tag{F.20}$$

where ρ is the traffic intensity and $E[s]$ is the service time, (s is constant) [73].

The Laplace-Stieltjes transform $B^*(s)$ of the service time for the $M/D/1/\infty$ queueing system is defined by

$$B^*(s) = e^{-sE[s]} \tag{F.21}$$

in [84] (p. 218), and the Laplace-Stieltjes transform $W_q^*(s)$ of the waiting time for the $M/D/1/\infty$ queueing system can be calculated from Equation (F.7) using Equation (F.21). Therefore, the Laplace-Stieltjes transform $W_q^*(s)$ of the waiting time in the queue for the $M/D/1/\infty$ queueing system is

$$W_q^*(s) = \frac{s(1-\lambda E[s])}{s - \lambda[1 - e^{-sE[s]}]}. \tag{F.22}$$

We can calculate the number of observations N required theoretically when estimating the mean waiting time in the $M/D/1/\infty$ queueing system with 5% of the relative statistical error for a 95% CI from

$$N = A(\rho)\left(\frac{1.96}{0.05 W_q}\right)^2, \tag{F.23}$$

where

$$A(\rho) = \sigma^2(W_q)\left[\frac{1+\rho}{1-\rho} + \frac{\lambda(W_q''' - W_q' W_q'')}{(1-\rho)(W_q'' - W_q' W_q')}\right], \tag{F.24}$$

(where $W_q^{'}$, $W_q^{''}$ and $W_q^{'''}$ can be obtained by the first, second and third differentiations of the Laplace-Stieltjes transform $W_q^*(s)$, in Equation (F.22), of the waiting times in the queue, respectively, and $\sigma^2(W_q)$ can be calculated by $(W_q^{''} - W_q^{'}W_q^{'})$); see [23] and [32].

The numbers of observations required in theory, with a relative statistical error of 5% and 10% at a 95% CI, are presented in Table F.2.

Table F.2: Required run-length when estimating the mean waiting time in the $M/D/1/\infty$ queueing system at a 95% CI

ρ	*Relative Statistical Rrror = 5%*	*Relative Statistical Rrror = 10%*
0.1	26,559	6,639
0.2	17,607	4,401
0.3	16,028	4,007
0.4	17,073	4,268
0.5	20,488	5,122
0.6	27,744	6,936
0.7	43,904	10,976
0.8	89,637	22,409
0.9	330,093	82,523

F.3 Run-Length for the $M/H_2/1/\infty$ Queueing System

The required observations in theory for a sequential steady-state simulation when estimating the mean waiting time for the $M/H_2/1/\infty$ queueing system can be also obtained in the same way.

The Laplace-Stieltjes transform $B^*(s)$ of the service time for the $M/H_2/1/\infty$ queueing system is defined by

$$B^*(s) = \frac{\alpha_1\mu_1}{\mu_1 + s} + \frac{\alpha_2\mu_2}{\mu_2 + s} \tag{F.25}$$

in [84] (p. 141), and the Laplace-Stieltjes transform $W_q^*(s)$ of the waiting time can be calculated from Equation (F.7) using Equation (F.25). Therefore, the Laplace-Stieltjes transform $W_q^*(s)$ of the waiting time in the queue for the $M/H_2/1/\infty$ queueing system is

$$W_q^*(s) = \frac{s(1 - \lambda(\frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\mu_2}))}{s - \lambda[1 - (\frac{\alpha_1 \mu_1}{\mu_1 + s} + \frac{\alpha_2 \mu_2}{\mu_2 + s})]}. \tag{F.26}$$

We can calculate the number of observations N required theoretically when estimating the mean waiting time in the $M/H_2/1/\infty$ queueing system with 5% of the relative statistical error for a 95% CI by

$$N = A(\rho) \left(\frac{1.96}{0.05 W_q} \right)^2, \tag{F.27}$$

where

$$A(\rho) = \sigma^2(W_q) \left[\frac{1+\rho}{1-\rho} + \frac{\lambda(W_q^{'''} - W_q^{'} W_q^{''})}{(1-\rho)(W_q^{''} - W_q^{'} W_q^{'})} \right], \tag{F.28}$$

(where $W_q^{'}$, $W_q^{''}$ and $W_q^{'''}$ can be obtained by the first, second and third differentiations of the Laplace-Stieltjes transform $W_q^*(s)$, in Equation (F.26), of the waiting times in the queue, respectively, and $\sigma^2(W_q)$ can be calculated by $(W_q^{''} - W_q^{'} W_q^{'})$); see [23] and [32].

The numbers of observations required in theory, with a relative statistical error of 5% and 10% at a 95% CI, are presented in Table F.3. We assumed α_1 = 0.09175, α_2 = 1 - α_1, μ_1 = 0.18350, and μ_2 = 1.81650.

F.4 Theoretically Required Run-Length When Estimating the Mean Response Time

Following the same procedures described above for the mean waiting time in the queue, we can obtain the number of observations required in theory when estimating the mean response time from the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems by applying the Laplace-Stieltjes transform $B^*(s)$ of the service time for the $M/M/1/\infty$, $M/D/1/\ \infty$, and $M/H_2/1/\infty$

Table F.3: Required run-length when estimating the mean waiting time in the $M/H_2/1/\infty$ queueing system at a 95% CI

ρ	*Relative Statistical Rrror = 5%*	*Relative Statistical Rrror = 10%*
0.1	149,966	37,491
0.2	136,379	34,094
0.3	144,345	36,086
0.4	163,911	40,977
0.5	198,230	49,557
0.6	259,526	64,881
0.7	383,801	95,950
0.8	710,709	177,677
0.9	2,308,130	577,032

queueing systems into Equation (F.4), respectively. The number of observations required in theory when estimating the mean response time with a relative statistical error of 5% and 10% at a 95% CI are presented in Table F.4. (Note that *RSE* in the table means the relative statistical error.)

Table F.4: Required run-length when estimating the mean response time from the $M/M/1/\infty$, $M/D/1/\infty$, and $M/H_2/1/\infty$ queueing systems

ρ	$M/M/1/\infty$		$M/D/1/\infty$		$M/H_2/1/\infty$	
	RSE=5%	*RSE=10%*	*RSE=5%*	*RSE=10%*	*RSE=5%*	*RSE=10%*
0.1	2,636	659	85	21	23,704	5,926
0.2	4,225	1,056	277	69	42,313	10,578
0.3	6,616	1,654	667	166	64,271	16,067
0.4	10,415	2,603	1,451	362	92,143	23,035
0.5	16,903	4,225	3,073	768	131,385	32,846
0.6	29,196	7,299	6,695	1,673	194,522	48,630
0.7	56,514	14,128	15,996	3,999	316,490	79,122
0.8	136,760	34,190	47,123	11,780	632,608	158,152
0.9	582,386	145,596	242,230	60,557	2,187,886	546,971

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my two supervisors, Associate Professor Krzysztof Pawlikowski and Dr. Donald McNickle, who have consistently assisted and encouraged me. I would never have been able to complete this dissertation without their continuous support and encouragement. I am very fortunate to have the opportunity to work with great supervisors having the kindest intention.

I also thank Dr. Greg Ewing for generously sharing his broad knowledge of Akaroa-2 and Mrs. Jane McKenzie for proofreading a draft of my thesis a number of times. Thanks also go to all the staff members of Computer Science Department who, in a variety of ways, made my PhD study at the University of Canterbury a pleasant and memorable experience.

During the course of my PhD study, I was awarded the G. B. Battersby Scholarship in Computer Science, and I participated a number of conferences with partial financial supports from Postgraduate Conference Funding in Department of Computer Science, Dr. D. McNickle's research grant, the IEEE New Zealand South Section, and the Royal Society of New Zealand - Canterbury Branch. That made the period of my PhD study enjoyable both academically and in life experiences. I would like to thank all of you provided those financial supports.

Finally, I am deeply grateful to my parents, my brothers and sisters, and friends for their unconditional love and support. Joshua Jeong deserves very special thanks for his understanding and assistance throughout all the times of my PhD study.

Sincerely thank all of you.