Performance of Sequential Batching-based Methods of Output Data Analysis in Distributed Steady-state Stochastic Simulation

Vorgelegt von
Diplom-Ingenieur
Edjair de Souza Mota
aus Manaus

Von der Fakultät IV- Elektrotechnik und Informatik
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
Dr.-Ing.

genehmigte Dissertation

Promotionsausschuss:
Vorsitzender: Prof. Dr.-Ing. Günther Hommel
Berichter: Prof. Dr.-Ing. Adam Wolisz
Berichter: Prof. Dr. Krzysztof Pawlikowski

Tag der wissenschaftlichen Aussprache: 08. Mai 2002

Berlin 2002
D 83

To mom, daddy and brothers, for their inestimable effort and love, without which I would be nobody. To my lovely Sheilla, love of my life, for her patience and dedication. To my little Sofia, the world became more beautiful with her.

Abstract

This research studies issues related to sequential analysis of time series arising from stochastic simulation of dynamic systems. The focus of this research is on design, implementation and performance assessment of confidence interval procedures based on batching methods, that should be run under an environment of multiple replications in parallel.

Investigation of properties of steady-state estimators continues being one of the most challenging enterprises. Necessity of design of efficient procedures for constructing confidence intervals of steady-state mean values is emphasized. When carefully implemented, the classical method of Nonoverlapping Batch Means (NOBM) works acceptably for simulation of queuing systems moderately loaded. However, for high-loaded system (which is frequently found in practice), the simulation practitioner needs more sophisticated methods to cop with strong serial correlation of observations arising from steady-state stochastic simulation, but care must be taken, so that the simulation run time should not be lengthen, and user attention should not focused more on the method of analysis than in the simulation itself.

The main challenge of procedures based on batching is that of determining the batch size. Analytical results are in general asymptotic, and oriented to a single-processor environment.

Nevertheless, one can take advantage of the availability of more (independent) processing power, and try to compensate an imprecise choice of batch size with higher degree of parallelization, by applying the approach of multiple replications in parallel, which gives us a chance of producing accurate results, especially when simulation on a single machine would be unrealistically long (so it would need to be stopped early, with inaccurate results).

This commitment is only possible when the procedure is robust and presents good statistical properties. Some sequential procedures were proposed, and carefully implemented aiming to fulfill both requirements. Empirical investigation showed us that they can give an attractive speedup, and at the same time to guarantee the accuracy of the final results. Statistical properties of the mean value estimators in such distributed environment are presented. As a case study, we investigated a wireless communication performance issue, in order to emphasize the feasibility of applying the proposed procedures here investigated to such complex dynamic system.

ACKNOWLEDGMENTS

I would like to thank God for my life and for the opportunity of knowing so interesting people. Thanks to Mr. Hercules Pimentel, who opened the doors for my MSc. and PhD. studies. Thanks to Prof. Dr. Fátima Turnell and Prof. Dr. Marcos Brasileiro, who introduced me to the challenging field of researching. Thanks to Prof. Dr.-Ing. Wolisz, for his careful and patient supervision and many moments of high level discussions which led me to an important conclusion : Knowledge without friendship looks like someone thirsty in front of the sea. Thanks to Prof. Dr. Pawlikowski for his technical support that let me see that knowledge without experience is fragile. Thanks to the staff of the Telecommunication Networks Group, that by means of professionalism let me perceive the way I should track to achieve my goals as a researcher. Special thanks to Rafael and Vânia Peregrino, Edson Silva, Frank Fitzek, Lothar Monthua and Petra Glaw for their friendships and moments of fun. Iza, Wolfgang, Ângela, Jocélia, Cherry, Augusta, Liana, Werner, Ana Lídia, Christian, Stella, great part of this effort belongs to you. At last but not least, I would like to thank the Brazilian people that by means of their taxes allowed our Government to send me to so advanced country to acquire knowledge, culture and experience that I must spread back all over the places I go.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Motivation

Nowadays we witness a great challenge in the computing area. As the communication technologies advance and computer processing power increases, new kind of applications arise, leading to an increasing demand for mobile services, providing user with access to electronic data and services "anytime and anywhere". Success in the development of complex wireless networks is partially related to the ability of predicting their performance already in the design phase and subsequent phases of the project as well. Terms like higher reliability, better coverage and services, higher capacity and mobile management, to say a few, are important issues to be included in the analysis.

Dynamic increasing of the complexity of such networks and the growth of the number of users require efficient tools for analyzing and improving their performance. Analytical methods of analysis are neither general nor detailed enough, and in order to get tractability, they need sometimes to make assumptions that require experimental validation. On the other hand, simulation, formerly known as *a last resort method* [83], is a flexible and powerful tool adequate for prototyping such complex systems.

To factors that have additionally stimulated the use of simulation, one could include: faster processors, larger-memory machines and trends in hardware developments (e.g. massively parallel processors, and clusters of distributed workstations).

Straightforward simulation of complex systems, e.g. communication networks, takes frequently a amount of computer time to obtain statistically valid estimates, despite increasing processing speed of modern computers. Ayani et al. [10] refer to a simulation experiment of a high speed local area network where the main objective was to estimate the setup delay and blocking probability of the control subnetwork; the execution time corresponding to one second of real time took around two hours.

Fitzek et al. [43] investigated the influence of jitter on the quality of service offered by a wireless link, and reported a simulation time as long as 180 hours using just 9 wireless terminals, though simulation has been executed in a fast workstation dedicated to that purpose.

Such phenomenon results from the statistical nature of the simulation experiments. Most simulation models contain stochastic input variables, and, thereby, stochastic output variables, the last ones being used for estimating the characteristics of the performance parameters of the simulated system. In order to obtain an accurate estimate with known (small) statistical error, it is necessary to collect and analyze sequentially a substantial number of simulation output data, and this can require a long simulation run.

Efficient statistical tools can be used to impact the running time of an algorithm by choosing an estimator with substantially lower computational demand. It would be a mistake to think that more processing power can replace the necessity for such tools, since the associated pitfalls can be magnified as well [56].

The need for effective statistical methods to analyze output data from discrete event simulation has concerned simulation users as early as 1963 [23]. Development of accurate methods of statistical analysis of simulation output data has attracted a considerable scientific interest and effort.

Even though, credibility of stochastic simulation has been questioned when applied to practical problems, mainly due to the application of not robust methodology for simulation projects, which should comprise at least :

1. The correct definition of the problem.
2. An accurate design of the conceptual model.
3. The formulation of inputs, assumptions, and processes definition.
4. Build of a valid and verified model.
5. Design of experiments.
6. Proper analysis of the output data.

Pawlikowski et al. [100] make a deep analysis on pitfalls of simulation output analysis, and suggest guidelines to overcome this fact.

Most of the statistical methods to analyze simulation output data were originally conceived for a single processor environment, and little is known about their behavior under a parallel computing environment. This is precisely the main focus of this thesis.

1.2 Research problem and hypothesis issues

We are concerned with stochastic simulation of processes that can be modeled as covariance stationary, that is, the first and second moments are finite over time and the covariance does not vary with absolute point of time t, but only with the distance between observations [74]. Some critical problems found in this scenario are :

1. Minimization of bias of steady-state estimates caused by initial conditions
2. Estimation of the sample variance of a performance measure and its confidence interval in the case of correlated observations in equilibrium state;
3. Stopping the simulation within a desired precision.

Except for regenerative simulations, data collected during transient phase are not representative of the actual average values of the parameters being simulated, and cannot be used to produce good estimates of steady-state parameters. The determination of its length is a challenging investigation, but it is out of the scope of this thesis. We assume that extension of the transient phase is conveniently detected by means of tests of stationarity such as those proposed in [118] [121], otherwise all results would be biased. Results obtained using a single processor confirmed the theory and, therefore, our assumption. Moreover, we assume that the stochastic process being analyzed converges to an equilibrium state.

Theoretically, steady-state occurs in the limit when the run length increases to infinity, but for practical purposes there is some point beyond which one can neglect the error that is made by considering the system to be in equilibrium. Underestimation of the length of the transient phase leads to bias in the final estimate. Overestimation, on the other hand, throws away information on the steady state and this increases the variance of the estimator [73].

Another difficulty is the nature of the output observations of a simulation model. Observations collected during typical stochastic simulations are usually strongly correlated, and the classical settings for assessing the sample variance can not be applied directly. Neglecting the existence of statistical correlation can result in excessively optimistic confidence intervals. For a thorough treatment of this and related questions refer to [99].

Construction of confidence intervals for steady-state mean values of simulated stochastic processes has been an attractive research area. Steady-state is by far the most common approach found in the literature of simulation output analysis, though it presents more methodological difficulties, and it requires proper statistical analysis of its properties, especially in the scenario of distributed simulation.

Investigation of properties of steady-state estimators continues being the most challenging enterprise, especially in our days when we witness an increasingly interest in using such complex dynamic systems as modern communication networks. Besides, it is a "common denominator" for comparing performance of various systems.

The ultimate objective of run length control is to terminate the simulation as soon as the desired precision of relative width of confidence interval is achieved. There is

a natural trade-off since one needs a reasonable amount of data to get the desired accuracy, but on the other hand this can lengthen the completion time. Considering that early stopping leads to inaccurate results, it is mandatory to decrease the computational demand of simulating steady-state parameters.

In order to decrease computational demands of intensive stochastic simulation one can (i) attempt to design a simulation that does not require the generation of a large number of observations; it is usually achieved by means of variance reduction techniques (VRTs), which try to increase the statistical efficiency of a simulation experiment by using statistical methods to reduce variance of estimators of performance measures or; (ii) dedicate more resources to the simulation experiment by means of parallel computing.

VRTs are experimental design techniques that improve the efficiency of a simulation experiment by reducing the cost of achieving an estimator with a specified level of precision. Although VRTs have been applied successfully in many cases (e.g. Izydorczyk [69], Frost [45], Devetsikiotis [28], Andreassen et al. [7], L'Ecuyer [84], Shang et al. [20]), in general they are model dependent and require good statistical background from the analyst. Moreover, a VRT applied inappropriately may even increase variance. Automation of variance reduction in simulation languages is supposed to be the only possibility for widespread applications of VRTs [8], but this has not been achieved at the time of this writing.

Parallel simulation is an attempt to speedup execution of simulation models by dedicating more computing resources for simulation and that constitutes a challenging problem which has attracted the attention of researchers as early as in [90]. Aiming to improve the performance of simulation experiments one can either:

1. divide a simulation model in submodels (or split the whole simulation task in functional units) and execute each of them on different processors; or

2. run the same model on several processors and average the results at the end.

The former approach is generally known as **SRIP** – **S**ingle **R**eplication **I**n **P**arallel, and the latter as **MRIP** –**M**ultiple **R**eplications **I**n **P**arallel (Pawlikowski et al. [106]). We lay emphasis on the latter strategy.

With the advent of massively parallel computers, where hundreds or even thousands of processors are connected together in a single computer to provide up to teraflop computing power, parallel simulation can potentially reduce simulation execution time by several orders of magnitude. A major challenge in running simulations on a massively parallel computer is to exploit sufficient parallelism for a large number of processors [129]. Though it was not the kind of environment we used, the results found in this research can be extended for statistical inference of stochastic simulation under massively parallel computers.

The problem of interest is to parallelize the simulation so that a large number of processors can be used to execute the simulation efficiently while producing an accurate estimate for the parameter of interest.

We are going to show in chapter 5 an investigation on how methods of analysis based on batching procedures are good in the sense of speedup when they are run

under **MRIP**. Those implementations were tuned, and we assessed the maximum number of processors (in the sense of the investigated scenario) that can be put to work together under **MRIP** according to the truncated Amdahl's law proposed by Pawlikowski and McNickle [102].

We are going to present important statistical properties of these sequential batching procedures, at first using one processor and then to investigate improvements offered by **MRIP**. Experimental analysis of coverage is required for assessing the quality of practical implementations of methods used for determining confidence intervals in steady-state simulation.

With many processors, one may do a distributed simulation of one long run or do independent replications with one replication per processor. In [132], it has been claimed that

> "*if the total simulation run length is long enough to obtain reasonably good estimates, then several independent replications are usually just as efficient as one longer run.*"

We hypothesize (H_1) that generating observations simultaneously by applying multiple independent replications in parallel, improves the efficiency of a long run-length simulation experiment, provided some careful procedures concerning statistical aspects have been taken into account. Therefore, as an attempt to partially fulfill the gap in the literature concerning statistical simulation output analysis under multiple independent replications in parallel, this research investigated a class of sequential methods based on batching of observations, that can be applied concurrently on workstations connected via a network.

We assume that the output process is weakly dependent (ϕ-mixing), meaning that its distant future behavior is essentially independent of its present or past behavior. A broad class of dependent stochastic processes possesses this ϕ-mixing property [19].

Therefore, an important issue with regard to stochastic simulation output analysis in distributed simulation, concerns the possibility of making use of more computing power aiming to achieve more accuracy in a shorter time interval. The generalization of methods of analysis found in the literature, to be run under multiple replications in parallel, yields new statistical properties that deserve deep investigation.

Batching is a classical methodology commonly used in simulation output analysis. A fundamental problem f batching methods is to find (1) a batch size large enough so that batch means may be considered independent, and (2) a number of batches large enough such that adequate degrees of freedom are provided to estimate the variance of the sample mean [70]. Research into automating this choice are supposed to be either "ad hoc in nature and lack a rigorous foundation" [21], or "too complicated to be implemented for practical use" [19]. We hypothesize (H_2) that by running batching methods under **MRIP**, one can compensate an imprecise batch selection by increasing the degree of parallelization, provided that candidate methods are consistent and robust.

In this research, we considered a method as being robust, when it demonstrated an ability to recover gracefully from the deviations of its underlying assumptions, and still produced results with high quality. A method of analysis is said to be consistent when the half-width of the confidence interval it produced tends to zero as the run length increases.

Classical Batch Means approach is based on fixed number of batches, but Glynn and Whitt [58] demonstrated the necessity of increasing the number of batches. After selecting the batch size, the procedure should test periodically the stopping conditions. We hypothesize (H_3) that after failing this test, the number of batches should increase slightly, to keep close to the range suggested by Schmeiser ($10 \leq b \leq 30$) [114].

When the process being investigated has high autocorrelation, it may be preferable to subsample the sequence of observations at a regular spacing **s**, making a new process with the same equilibrium distribution but less autocorrelation [50]. We hypothesize (H_4) that the autocovariances provide a useful guide to selecting the spacing among batches and the more suitable spacing for each replication can be found by calculating the correlation coefficient of lag 1 for the first batch.

Another family of output analysis methods we investigated is the *standardized time series method has asymptotic* (STS), which has been proved to offer asymptotic advantages over the batch-means based methods [62], but Sargent et al. [113] detected that it can require very long runs. We hypothesize (H_5) that by applying this class of methods under **MRIP** we can get rid of this kind of difficulty by generating observations more quickly, making STS is very attractive for sequential analysis of stochastic simulation, especially when simulation output observations are highly correlated.

1.3 Justification for the research

The last two decades can be cited as an example of increasing the importance of simulation output area as one examines a plethora of different techniques developed and investigated by researchers for drawing conclusions from simulation experiments. As pointed out in a panel discussion of the 1994 Winter Simulation Conference [18], many of these techniques are hardly used in practice mainly because they *"require extensive academic background in esoteric subjects merely to understand them"* (Carson).

That claim emphasizes a practical problem, but at the same time gives orientation for further research : Efficient tools for automatically analyzing simulation output data should be based on secure and robust methods that can be broadly and safely applied to a wide range of models without requiring from simulation practitioners highly specialized knowledge.

The research problem presented here has received relative few interest by previous researchers because the most research effort on parallel simulation has been directed

toward experiments based on SRIP. Besides this focus point, the methodology for assessing the applicability of new implementations should be somewhat revisited. In section 1.5 we propose a combination of robust methodologies to obtain a more general methodology suitable for that purpose.

Finally, results obtained in this research has been applied successfully to complex computer communication networks problems, reducing the usually prohibitive very long run time, and concomitantly assuring high level of accuracy.

1.4 Goals of this thesis

Several methods have been proposed in the literature to assess the credibility of the estimation of the steady-state mean of a performance parameter, however, how these methods should perform in a parallel environment, where observations are generated in an asynchronous and distributed way, deserves rigorous investigation.

Primary goal of using parallel environment is to reduce the simulation execution time in order to simulate a great amount of events. How should that be carried out ? Speedup (defined for each number P of processors as the ratio of the elapsed time when executing a program on a single processor to the execution time when P processors are cooperating to execute the program) and efficiency (defined as the average utilization of the n allocated processors), though traditional performance criteria, are not enough to describe the behavior of a parallel system [30]. We associated these metrics to the quality of the simulation results, since it would be of little worth to obtain wrong results with a sound speedup .

Very little is known about quality of output analysis methods in sequential simulations and in fast concurrent sequential simulations based on **MRIP**, when processors cooperate in production of data. Practical results can be found in [86] [94] [95] [104].

This work is concerned with a methodology of automated sequential simulation output analysis with emphasis on parallel simulation scenario provided by multiple replications in parallel time streams approach, with particular attention to sequential confidence interval procedures based on data batching techniques.

Through investigation of some batching-based simulation output analysis techniques, originally conceived for a single processor scenario, we proposed, designed and implemented sequential confidence interval procedures (CIPs) based on these techniques, in such a way that they can be run under **MRIP** scenario. Analysis and performance evaluation were carried out as well. We intend to identify the best sequential procedure strategy, among those we have analyzed, as well as provide additional insight into the trade-off between speedup and accuracy of the results.

Since sequential analysis of simulation output data is the only effective way of controlling the statistical error of the final results, we propose sequential procedures that are as efficient as possible, both in the sense of their speedup and accuracy in the **MRIP** scenario.

Our purpose is to contribute to the methodology for automated sequential stochastic simulation with emphasis on parallel environment provided by Akaroa-2, a simulation controller designed at the Department of Computer Science of The University of Canterbury, Christchurch, New Zealand. There are many problems which need to be solved before such methodology could be commonly used. To my best knowledge nobody has done exhaustive analysis of coverage (as we did) for the attractive methods investigated in this research, namely *spaced batch means*, *overlapping batch means* and *standardized time series*. Additionally, we also consider as a contribution an up-to-day survey of the methods of simulation output data analysis.

The novelty of this research is the validation of this class of steady-state simulation output analysis concerning their feasibility under **MRIP** scenario, which gives rise to very challenging problems due parallelization of computing processes.

1.5 Methodology

The cornerstone component of the methodology adopted in this research is based on the conceptual framework proposed by Schriber and Andrews [116] for constructing confidence interval.

The main measure of effectiveness of methods of analysis is the coverage of the results, which traditionally has been assessed in a fixed-sample-size fashion. We adopted the sequential coverage analysis proposed in [104], since it is more reliable and suitable for robustness issue we aim, and based on the survey in [85] we constructed the coverage function proposed in [117].

Looking for a wide-range applicability, we combined the above approaches to find a reliable methodology to investigate sequential batching procedures under MRIP. By applying this methodology, we can additionally assess both the sample size required by each designed, implemented and investigated sequential procedure, as well as the accuracy of them, measured by the confidence interval half-width.

After obtaining promising confidence interval procedures that could be safely used under **MRIP**, we investigated the influence of the degree of parallelization over the overall performance. For sake of tuning, we checked the robustness of these CIPs. It consisted of violating deliberately the assumptions these promising procedures are based upon, and gradually increase the degree of parallelization to compensate the negative effects for the not fulfillment of the assumptions. As a matter of fact, we applied this investigation for every CIP, but only two of them presented very attractive performance.

In the sequence, we investigated alternatives to assess the achievable speedup by each CIP by tuning checkpoints granularity. For this purpose, we adopted the methodology proposed by Ruth Lee et al. [87].

1.6 General overview of the results

We initiated our investigation reproducing very-well known results concerning the classical nonoverlapping batch means (NOBM) in the context of a single-processor environment. These results were used as a reference to our investigations, but we went a step further and collected informations frequently just cited in the literature, namely, performance of sequential analysis procedures under high level traffic intensities.

Despite using the best choices for each issue of a stochastic simulation (e.g., reliable random numbers, sophisticated detection of run length, and non-parametric method for the complex problem of estimating correlation coefficients of a dependent time series), it was not possible to improve the coverage of the classical nonoverlapping batch means under high level traffic intensities, even adding more processors.

We suggested alternative procedures to overcome the above difficulties, and implemented sequential versions to be run under Akaroa-2, an efficient environment of multiple replications in parallel (**MRIP**) : Nonoverlapping batch means with increasing number of batches (NOBM/GW), and spaced batch means (SBM). Empirical investigation confirmed they are indeed improvement over NOBM, but they do not achieved acceptable level of coverages for high traffic intensities. Moreover, they offer a more important obstacle to be overcome : in the estimation phase, checkpoints must occur only in distances multiple of batch size. This will certainly lengthen the non-parallelizable part of a simulation, avoiding them to be used under **MRIP** when speedup is an issue.

Overlapping batch means (OBM) was our next choice, and due its excellent statistical properties, together with the fact that each observation can initiate a new (overlapped) batch, we have at our disposal a powerful procedure that can give us coverage close to nominal level even under high traffic intensities. Additionally, sequential version of OBM permits us to perform fine granularity tuning of checkpoints, yielding sound results on speedup.

Two sequential procedures based on the classical-sum of standardized time series proved to be very robust in the sense of all measures of efficiency. In one version (CSUM.1), checkpoints occur after collecting a number of observations equal to the selected batch size, the same way as in NOBM. In the second version (CSUM.2), checkpoints occurs according to a fine granularity. At least under **MRIP** both are not suitable for stochastic simulation of queuing systems with low/medium traffic intensity. On the other hand, CSUM.2 performed very well in the sense of speeding up simulation experiments.

1.7 Thesis Outline

This document is organized as follows.

Chapter 2 reviews some concepts on simulation output analysis and parallel simulation, necessary for the understanding the main problems of our research.

Chapter 3 describes the methodology used to conduct properly the research explained in this work.

Chapter 4 was dedicated to issues concerning design of sequential confidence interval procedures based on batching techniques to be run under multiple replications in parallel, their properties and assumptions, as well as the decisions made to answer the main issues of each analytical method underlying the procedures.

Chapter 5 contains performance evaluation results obtained from exhaustive studies of these sequential procedures over a range of reference queuing systems.

Chapter 6 presents a case study on a wireless communication performance issue, in order to emphasize the feasibility of applying the proposed procedures under MRIP.

Chapter 7 summarizes our results, as well as points out directions for future research.

2

Literature review

2.1 Introduction

Chapter 1 situated this research inside a wider body of knowledge, namely the sequential confidence interval procedures for assessing the quality of simulation results. This chapter reviews the necessary background and literature with respect to quantitative stochastic simulation, sequential methods of analysis, and parallel simulation for estimating the steady state mean of a stochastic process.

2.2 Basic Concepts of Quantitative Stochastic Simulation

Stochastic simulation is essentially a controlled statistical sampling technique commonly used by scientists and engineers to investigate complex probabilistic problems. It is comparable to analysis by experimentation, since one comes frequently across the usual problems associated with running experiments in order to make inferences about real systems, and must be concerned with such things as run length, number of replications, and statistical significance [63].

Any quantitative stochastic simulation study should include proper statistical analysis of output data collected during the simulation, otherwise, despite careful efforts on verification, validation, and selection of a good generator of pseudorandom numbers, there would be no guarantee of an acceptable level of quality of the results. As stated by J. Kleijnen [72],

> "... computer runs yield a mass of data but this mass may turn into a mess (...) if the random nature of output data is ignored and instead of an expensive simulation model, a toss of the coin had better be used".

Suppose we are investigating a steady-state measure of performance θ (e.g. the mean waiting time in a queue) by means of a simulation experiment. For this purpose :

1. Let $\{X_i\}$ be realizations of simulation output stochastic process that converges in distribution to a random variable $\mathbf{X}$.
2. We should detect the amount of observations subject to initialization bias[1], and discard them.
3. We can estimate the sample mean by calculating the arithmetic average of the sample

$$\bar{X}(n) \equiv \frac{1}{n}\sum_{1}^{n} X_i \tag{2.1}$$

where n is the sample size (the run length minus the number of initial observations discarded during the transient phase).

To assess how close the estimator of the mean is to θ, we construct a confidence interval using the *Central Limit Theorem*, which states that if $\{X_i\}$ are independent and identically distributed with finite mean and variance[2] and meet other relatively weak conditions, then $\bar{X}(n)$ has approximately the normal distribution when n is large, regardless the distribution of $\{X_i\}$. This convergence in distribution is achieved as $n \to \infty$, which means that this interval estimation is an approximation.

Let approximate $100(1-\alpha)\%$ confidence interval estimator I for the mean θ of the underlying process be :

$$I : \mu \in \bar{X}(n) \pm H \tag{2.2}$$

where H is the half-width of this confidence interval, estimated as

$$H = t_{d_f,1-\alpha/2}\sqrt{Var\{\bar{X}(n)\}}, \tag{2.3}$$

[1] Except for the method of regenerative simulation, where the initial transient problem doesn't need to be identified.

[2] By assuming that $\sigma^2 < \infty$ and that $\{X_i\}$ are independent and identically distributed (IID), the Central Limit Theorem cannot be used to estimate the accuracy of the means of some self-similar processes, which have infinite variance and, therefore, out of the scope of this research. Readers interested can look for limiting theorems for heavy tailed random variables first formulated by Lévy (Feller [34], Samorodnitsky and Taqqu [111])

$t_{d_f,1-\alpha/2}$ is the upper $(1-\alpha/2)$ critical point of the t-distribution with d_f degrees of freedom, and $Var\{\bar{X}(n)\}$ is an estimator for the variance of sample mean, defined as

$$Var\{\bar{X}(n)\} \equiv \frac{\sigma^2}{n} \tag{2.4}$$

Taken together, the point estimate $\bar{X}(n)$ and the confidence interval I say what is the best guess for μ, and how far in error that guess might reasonable be. As σ^2 is usually unknown, one uses $S^2(n)$, the sample variance estimator of σ^2, given by

$$S^2(n) = \frac{1}{(n-1)} \sum_{i=1}^{n} \{X_i - \bar{X}(n)\}^2 \tag{2.5}$$

When $\{X_i\}$ form a covariance stationary process, Fishman [37] demonstrated that

$$Var\{\bar{X}(n)\} \equiv \frac{\sigma^2}{n}[1 + 2\sum_{j=1}^{n-1}(1 - j/n)\rho_j] \tag{2.6}$$

where ρ_j is the correlation coefficient at lag j.

Equation 2.6 points out at the main difficulty found in most stochastic simulation. Namely, the fact that output observations never form independent and identically distributed normal output [60], but usually highly correlated, causes that straightforward analysis of $\mathrm{Var}\{\bar{X}(n)\}$ by classical statistical techniques is not possible, and ignoring autocorrelations is unacceptable, since the reliability of the sample mean and sample variance could be strongly overestimated [35].

2.3 Confidence Interval Procedures

To solve this tactical problem, several methodologies for estimation of confidence intervals of the mean of a sequence of correlated observations have been proposed. They include : *independent replications* [36], *regenerative approach* [26], *batch means* [23], *spectral analysis* [67], *standardized time series* [119], to say a few. Each of these ideas has its own strengths and weaknesses. The main difference among them is the way they estimate $Var\{\bar{X}(n)\}$. Efficient implementations of confidence interval procedures (**CIP**) and corresponding performance under **MRIP** should be carefully quantified, as Bischak et al. [15] observed, *"there is no procedure that, by general consensus, is preferred in all simulation experiments"*. A thorough review of these methods can be found in Pawlikowski's survey [99].

It raises the question of which of them should be used during a given simulation experiment. How do these methods differ ? Systems presenting regeneration cycles, though interesting, are not considered in this text. Readers interested on this kind of simulation can find details on this approach in [24] [25] [38] [54] [53] [52] [55]. A sequential version of the method Spectral Analysis under **MRIP** has been exhaustively investigated in, e.g., [105] [108] [106].

Batch means continues being the most widely used method for statistically analyzing simulation output, as it is simple to implement and understand. Moreover, an estimator based on batch means possesses some other desired properties [61]:

- Except for estimating the optimal batch size, defined here as the minimum number of observations per batch that can yield almost uncorrelated batch means, it is a method of ease computation.
- It does not require large amount of storage.
- Tt can be easily incorporated into procedures of simulation output analysis.

Batch means approach attempts to work around the correlation structure by rearranging the overall data into subsets which have uncorrelated batch means. Long time ago, Student (1927) wrote : "*A number of determinations of the same thing made on the same day are likely to lie mode closely together than if repetitions has been made on different days*". Brillinger [17] formalized this intuitive reasoning, namely, observations far apart from each other are less correlated than if they were closer. That is precisely the idea behind batch means, that is, above certain length m^* of batch size, batch means are (almost) independent. Thus, determination of batch size is the main issue of this method.

In the next section we are going to present some variants of the method of *batch means*, and some variants of the method of *standardized time series*. They are the basis for sequential versions we investigated under **MRIP**.

2.3.1 NOBM

In the classical setting, hereafter called *nonoverlapping batch means* (**NOBM**), a series of steady-state observations $\{x_i\}$ of length n is divided into b contiguous batches of size m, and the mean of the the j^{th} batch can be found through

$$\bar{X}_j(m) = \frac{1}{m} \sum_{i=m(j-1)+1}^{mj} x_i, \ \ for j = 1, \ldots, b; \tag{2.7}$$

For large enough batch size m^*, the dependency and the nonnormality of the batch means $\bar{X}_j(m)$ are negligible, with error that diminishes as m and n approaches infinity. Equivalently, the correlation among the batch means diminishes as m and n approaches infinity [3].

The overall point estimator is given by

$$\bar{X}(n) = \frac{1}{b}\sum_{j=1}^{b}\bar{X}_j(m) \tag{2.8}$$

Thus, a confidence interval can be estimated by means of

$$\bar{X}(n) \pm t_{d_f,1-\alpha/2}\sqrt{\frac{S^2(b)}{b}} \tag{2.9}$$

where $d_f = b - 1$ and

$$S^2(b) = \frac{1}{(b-1)}\sum_{j=1}^{b}(\bar{X}_j(m) - \bar{X}(n))^2 \tag{2.10}$$

A measure of the bias is given in [15] :

$$bias(b,m) \equiv \frac{b/[1 + 2\sum_{j=1}^{b-1}(1 - j/b)\rho_{\bar{x}}(j)] - 1}{b-1} \tag{2.11}$$

where $\rho_{\bar{x}}(j)$ is the correlation of lag j of the batch means.

Positively correlated output streams occur more frequently than negatively correlated output [112] [74] [130]. From (2.11), positive $\rho_{\bar{x}}$'s result in $b(k,m) < 1$, which denotes negative bias, and the final confidence interval half-width in Equation 2.9 can be underestimated, which leads frequently to a final coverage less than the nominal confidence level. Coverage is defined as the relative frequency that the final confidence intervals contain the true parameter θ.

The selection of the batch size will have direct influence on the variance estimator, and, therefore, the quality of estimation, measured by the coverage of the results. There is no foolproof recipe for deciding how big a batch should be. Considering a sample size n, large batch size guarantees independence and normality among batch means, besides assuring good coverage, but there is evidently loss of information, since the expected half-width of the confidence interval is very large, and it can lead to problems with the application of the central limit theorem, as there are fewer batches available. On the other hand, short batch sizes put in check normality among batch means, as the batch means can be highly correlated.

Schmeiser [114] conducted an extensive quantification study on effects of choosing batch size. By using a fixed sample size, he investigated theoretically the behavior

of the half-width of a confidence interval, considering its expected value, $E\{H\}$, its variance, $V\{H\}$, and its coefficient of variation, $CV\{H\} = \sqrt{V\{H\}}/E\{H\}$, and the probability that the interval covers all points $\theta_1 \neq \theta$ (the lower this probability the better the procedure, since it is equivalent to Type II error), for different batch sizes at different confidence level.

Schmeiser's studies pointed out that less than 10 batches yields in confidence intervals highly variable, and additional batches beyond 30 yield no improvement in the sense of accuracy of the expected half-width of confidence intervals, though they result in a smaller variance of the half-width.

We emphasize our interest in sequential analysis of steady-state mean values from simulation output data, but the important result of Schmeiser, though derived for a fixed sample size, will be taken into account during design and performance analysis of the sequential procedures developed to that purpose.

2.3.2 NOBM/GW

Schmeiser's results [114] for fixed sample size are based on the fact that low degrees of freedom result in small bias and large variance [115].

Keeping the number of batches fixed, though, imposes an asymptotic validity problem, as

> *"there does not exist a batch-means estimation procedure based on a fixed number of batches that is consistent"* [58].

Therefore, as the run length increases, the number of batches should increase as well. In terms of sequential analysis it means that whenever the **CIP** needs to collect more observations to acquire a specific relative precision for stopping the simulation experiment, it should increase the size of batches in order to keep the number of batches within Schmeiser's proposed range. But how much should be this increment ?

After failing the test against relative precision for the b batch means suggested by Schmeiser ($10 \leq b \leq 30$), the number of batches should increase slightly, that is, not far from this suggested range. This procedure will be called hereafter **NOBM/GW**.

2.3.3 SBM

As the main analytical problem in terms of steady-state simulation is the usually high correlation among observations, one should look for manners to mitigate this negative effect, for instance, by creating spaces among the output sequence $\{X_i\}$ at regular distance, e.g. by discarding the last s observations of each batch. The resulting process has the same equilibrium distribution but less autocorrelation [50].

Conway [23] suggested the use of a fixed spacing regardless the sample size. Billingsley (1986) showed that by breaking the observations into alternating long batches of length $\lfloor n^{3/4} \rfloor$ and small spaces of length $\lfloor n^{1/4} \rfloor$ where $\lfloor . \rfloor$ is the greatest integer function, the spaced batch means are asymptotically IID [44].

Fox et. al [44] proposed discarding some observations between contiguous batches, and called this approach *Spaced Batch Means* (**SBM**). This kind of subsampling attenuates the serial correlations among the observations, by producing another process with less observations per batch, **s** observations apart from each other.

Considering that the first $m - s$ observations of each batch are incorporated into the respective batch means, the j^{th} spaced batch mean is defined by

$$\bar{X}_j(m-s) = \frac{1}{m-s} \sum_{k=1}^{m-s} X_{(j-1)m+k}, \ \ for j = 1, \ldots, b; \tag{2.12}$$

and the overall point and interval estimators are given by

$$\bar{X}(n) = \frac{1}{b} \sum_{j=1}^{b} \bar{X}_j(m-s) \tag{2.13}$$

A confidence interval can be estimated by means of

$$\bar{X}(n) \pm t_{d_f, 1-\alpha/2} \sqrt{\frac{S^2(b)}{b}} \tag{2.14}$$

where $d_f = b - 1$ and

$$S^2(b) = \frac{1}{(b-1)} \sum_{j=1}^{b} (\bar{X}_j(m-s) - \bar{X}(n))^2 \tag{2.15}$$

When the number of discarded observations **s**=0, we have the classical NOBM.

We claim that for high values of traffic intensity, letting (conveniently chosen) spaces between batches can give always better results, when compared to classical nonoverlapping batch means. The problem could appear in a conversely situation, that is, for low values of traffic intensity, spacing could be irrelevant, and even wasteful.

2.3.4 OBM

Meketon and Schmeiser [89] proposed a method known as *Overlapping Batch Means* (**OBM**) that considers it is more important to have a larger number of batches than to have these batches independent. Thus, for a given sample size, some observations from previous batches can be reused for generating more batch means. The number of degrees of freedom depends on the degree of batch overlapping, and is greater than the degrees of freedom in NOBM.

OBM has been introduced to increase the degrees of freedom of the estimator of the mean value used in NOBM. By grouping observations into overlapping batches, if one reuses all observations (except the first one) from the previous batch, one obtains the overlapping of batches (hereafter called complete OBM), which has degrees of freedom 1.5 times greater than the degrees of freedom in NOBM. Hence, one can use all $n - m + 1$ overlapped batches of size m to find $\bar{X}(n)$ and $\hat{V}(\bar{X}(n))$.

Let

$$\bar{X}_i(m) \equiv \frac{1}{m} \sum_{j=0}^{m-1} X_{i+j}, \tag{2.16}$$

be the i^{th} overlapping batch means, i = 1, ..., n-m+1.

The overall point estimator is given by

$$\bar{X}(n) = \frac{1}{n-m+1} \sum_{j=1}^{n-m+1} \bar{X}_j(m) \tag{2.17}$$

A confidence interval can be estimated by means of

$$\bar{X}(n) \pm t_{d_f,1-\alpha/2} \sqrt{\frac{S^2(n)}{b}} \tag{2.18}$$

where $d_f = 1.5(b-1)$ and

$$S^2(n) = \frac{1}{(n-m+1)} \sum_{j=1}^{n-m+1} (\bar{X}_j(m) - \bar{X}(n))^2 \tag{2.19}$$

Considering that the optimal batch size m^* is the same as found in NOBM, the variance of the sample mean by applying the complete OBM is less than 1/3 of variance of the sample mean by applying NOBM, as long as $n \to \infty$. Welch [131] concluded that with modest degree of overlapping (e.g. by m - m/4 observations), a satisfactory variance reduction can be achieved.

2.3.5 STS

Standardized Time Series (**STS**) is an approach that takes other order of ideas to generate confidence intervals for steady-state simulation, although it is still based on batching. Instead of standardizing a single scalar, e.g. the sample mean of an output time series X_i (i=1,2, ..., n), Schruben [119] suggested the standardization of each observation ot the time series, defined as

$$T_i(t) \equiv \frac{\lfloor mt \rfloor (\bar{X}_{i,m} - \bar{X}_{i,\lfloor mt \rfloor})}{\sigma\sqrt{m}},\ 0 \le t \le 1, \tag{2.20}$$

where

$\bar{X}_{i,j}$ is the cumulative average of the first j observations in the i^{th} batch
$\lfloor . \rfloor$ denotes the greatest integer function.

The transformed series converges asymptotically to a standard Brownian bridge process, which properties area used to construct a confidence interval.

After standardizing each observation one can find the random variables A_i, the asymptotic scaled sum of $T_i(t)$, for each batch by means of

$$\begin{aligned} A_i &= \sigma\sqrt{m}\sum_{k=1}^{m} T_i(t) \\ &= \sum_{k=1}^{m}\sum_{j=1}^{k}(\bar{X}_i - \bar{\bar{X}}) \end{aligned} \tag{2.21}$$

Schruben assumes that for large m the A_i are approximately normal IID, and proposed an asymptotically valid area interval estimator $I_A = [\bar{X}(n) \pm H_A]$ for the performance parameter θ, with half-width

$$H_A = t_{d_f,1-\alpha/2}\sqrt{12\sum_{i=1}^{b}\frac{A_i^2}{n^2(m^2-1)}}. \tag{2.22}$$

where $d_f = b$.

By computing the statistic

$$A = \sum_{i=1}^{b}\frac{12{A_i}^2}{(m^3-m)} + m(\bar{X}(n) - \bar{X}_i)^2 \tag{2.23}$$

Schruben also proposed an asymptotically valid combined *classical-sum* interval estimator $I_{csum} = [\bar{X}(n) \pm H_{csum}]$ for the performance parameter θ, considering that

$$H_{csum} = t_{d_f,1-\alpha/2}\sqrt{\frac{A}{n(2b-1)}}. \tag{2.24}$$

where $d_f = 2b - 1$.

One can note that STS-based estimators do not try to estimate the variance of the sample mean to construct the final confidence interval, but on the contrary its underlying principle is to cancel it out. Theoretical foundations of the method can be found in [57].

Though standardized time series method has asymptotic advantages over the batch means method [62], Sargent et al. [113] detected that it can require very long runs. We are going to see that by applying **STS** under **MRIP** we can get rid of that difficulty by generating observations more quickly.

The motivation to dedicate part of our research on this method was based on the following : (a) the simplicity of its implementation, though the underlying sophisticated statistical techniques; (b) as batch-means based procedures require some time to detect independence among batch means, STS requires some time to detect normality among A_i, but this preprocessing is in general shorter, since it requires

a smaller number of batches (e.g. 10 batches) to

test for normality. This characteristic signalizes that STS can be a promising method to be investigated under **MRIP** and its performance should be quantified. In this research we have investigated the *classical-sum* interval estimator.

2.4 Looking at the data sequentially

It is not possible to know in advance the sample size needed to meet the required precision of a given problem. The only effective way to control the statistical error of an estimation procedure is to analyze the data sequentially. An appropriate sequential stopping rule can be used to control the width of the estimated confidence interval.

The run length of a stochastic simulation experiment can be determined either by assigning the amount of simulation time before initiating the experiment or by letting the simulation run until a prescribed condition occurs. The first approach is known as fixed-sample size procedure, and suffers from the possibility of inappropriate precision of the results. The second approach is generally known as sequential procedure and is the subject of this research.

Sequential procedures gather output observations at pre-specified checkpoints to investigate a certain parameter of interest, and a decision has to been taken to make the required estimation and stop the sampling if a predefined condition is achieved, or to continue the sampling and periodically repeat both steps above while necessary. It is evident that the number of observations required to terminate the experiment is a random variable. Thus, a sequential procedure can be economical in the sense that a decision may be reached earlier when it is compared to fixed-sample-sized experiments, but can be onerous if one wishes a tight precision.

The importance of sequential procedure is widely recognized as the only effective method for controlling the precision of simulation results. Two fundamental issues that motivate the design of more efficient sequential procedures are the possibility of specifying the desired precision for the estimated parameter, and the termination rule to conclude the experiment whether the precision has been reached. Ideally, the rule should be computationally easy and not lengthen the completion time at all.

The general problem of sequential estimation is to postulate a sampling rule that will ensure that the unknown parameter is estimated with a given accuracy and with minimum expected sample size. The accuracy is usually measured in the form of the width of the confidence interval, which is a random variable itself and cannot be bounded if the sample size is fixed in advance.

2.4.1 Sequential procedures

A number of sequential procedures have been proposed for constructing confidence intervals for the mean of a sequence of independent observations of the random variable of interest. Chow and Robbins [22] presented a sequential confidence interval procedure for the unknown mean θ of a population with finite variance, by defining an "absolute accuracy". The final confidence interval should have a prescribed width $2d$ with probability $(1-\alpha)$, that is :

$$I_n = (\theta : |\bar{X}(n) - \theta| \leq d) \tag{2.25}$$

Simulation should stop for the smallest odd $n \geq 3$ for which

$$S^2(n) < n \left(\frac{d}{t_{n-1,1-\alpha/2}} \right)^2 \tag{2.26}$$

where $t_{n-1,1-\alpha/2}$ is the critical value of the t distribution for $1-\alpha/2$ and $n-1$ degrees of freedom.

They proved that as n increases, $d \to 0$ and the coverage closes to $\gamma = 1-\alpha$. This is known as asymptotic consistency, i.e. this sequential estimation procedure covers θ with probability $1-\alpha$. On the other hand, the magnitude of the parameter being

estimated is not always known in advance, and choosing d inadequately can produce unreliable results.

Nadas [96] extended Chow and Robbins' method by proposing a procedure to estimate θ to be within $100\gamma\%$ of $|\theta|$ with probability $1-\alpha$, the so called "proportional accuracy".

$$I_n = (\theta : |\bar{X}(n) - \theta| \leq \gamma|\theta|) \tag{2.27}$$

Let $v^2(n) = [1 + (n-1)S^2(n)]/n$. Simulation should stop for the smallest $n \geq 2$ for which

$$\frac{(t_{d_f,1-\alpha/2})v(n)}{\sqrt{n}\ |\bar{X}(n)|} \leq \gamma \tag{2.28}$$

where $d_f = n-1$. Law et al. [82] compared this procedure with a slightly modified one, which states that the half-width should not be more than $100\gamma\%$ of the magnitude of the point estimate. Let $d(n) = (t_{n-1,1-\alpha/2})S(n)/\sqrt{n}$. Simulation should stop for the smallest $n \geq 2$ for which

$$\frac{d(n)}{|\bar{X}(n)|} \leq \gamma \tag{2.29}$$

Both procedures were applied to twelve stochastic models for which analytical results are available, and computed the proportion of coverage and the average final sample size in each case. A 90% confidence interval was constructed, and 500 independent experiments were carried out. They concluded that choice between both procedures depends on the applied confidence level, but Nadas' proposal gave better coverage, but as they used a fixed number of experiments to assess the coverage, the quality of these results can be questionable.

Lavenberg and Sauer [80] proposed sequential stopping rules, which control the relative width of an estimated confidence interval to be used in conjunction with regenerative simulation. At least for the queuing systems they investigated by applying the procedures to construct a 90% confidence interval, a relative width of 0.05 was small enough to yield valid confidence intervals in almost all experiments. We adopted this relative width as the stopping rule for our empirical investigation.

The advantage of the relative width criterion for stopping simulation is that it allows the analyst to express the desired half length in terms of the mean point estimator. We look for a sequential procedure that can both determine the sample size during the course of the simulation run, and obtain confidence intervals with "acceptable" coverage.

A given sequence of observations gathered at the output of a simulation model is investigated at a certain time and a decision has to be taken to stop the sampling or

to continue the sampling. It means that the width of the confidence interval of an estimated parameter is compared against the assumed requirements at consecutive checkpoints of the simulation experiment, and a decision is made either to stop the simulation if the required accuracy is reached, or to continue the simulation for collecting the next portion of observations, unless the maximum allowed length of simulation run has been reached, otherwise.

Concerning *batch means*, Gross and Harris [63] suggested fixing the number of batches b and increasing the batch size m until the estimated correlation at lag 1 between $\bar{X}_i(m)$ (i=1,...,b), can be considered small enough. The difficulty is that correlation estimators are generally biased and for small number of batches they have a large variance.

Fishman [39] proposed a sequential batch means procedure based on von Neumann statistic [97]. Let $\{\bar{X}_i(m) : \text{i=1,2,...,}\ \text{b}\}$ be a sequence of batch means of size m for a certain amount of sample size n. Setting initially the batch size to $m = 1$, one should calculate the batch mean and the following statistic to test the level of correlation between the b batch means of size m :

$$C_b = 1 - \frac{\sum_{i=1}^{b-1}(\bar{X}_i(m) - \bar{X}_{i+1}(m))^2}{2\sum_{i=1}^{b}(\bar{X}_i(m) - \bar{X}(n))^2} \tag{2.30}$$

If $\bar{X}_i(m)$ are normal, then under null hypothesis H_o of independence, C_b has mean zero and variance $(b-2)/(b^2-1)$. For a specified quantile $(1-\beta)$ from the standard normal distribution, if $C_b > z_\beta\sqrt{(b-2)/(b^2-1)}$, then m=2m and after obtaining the sample size n, the von Neuman test should be repeated. F_{C_k}, the distribution of C_b, should become already close to normal for small number of batches (say $b \geq 8$). Kleijnen et al. [76] emphasized that the power of this test statistic is small if the number of batches is less than 100.

Law and Carson [81] proposed initially dividing the sample size n into lb batches (e.g. l=10, b=40) of size $m = 2$. If correlation at lag 1 $\rho_1 > c$, where c is a given threshold, the sample size ie doubled, otherwise n is divided into $lb/2$ batches of size $2m$, the new threshold is $c = \rho_1$. If the new $\rho_1 > c$, the procedure computes a confidence interval according to (2.2), otherwise, the sample size is doubled and so on.

Adam [1] developed the blocking method, a modification of Fishman's method to make it suitable for determining the simulation run length needed to meet a preassigned width of the confidence interval. For assessing the performance of his proposal, Adam simulated queuing systems and obtained how often the resultant 90% confidence intervals covered the true value, as well as the respective sample size and the relative half-width.

An interactive sequential confidence interval procedure based on standardized time series to produce results that satisfy a relative precision, was proposed by Duersch

and Schruben [29]. The user is asked to inform the largest period of time that can be spent in the experiment, but the procedure proceeds for only 20% of the that time limit. The user may accept or reject the intermediate time period. In the last case, the additional simulation time is estimated, and simulation goes on until acceptable results are obtained or the time limit is achieved.

Steiger and Wilson [125] proposed a batch-means procedure called ASAP (Automated Simulation Analysis Procedure). The procedure requires that the output process be ϕ-mixing a mild condition found virtually in all practical settings. Intuitively, a process $\{X_i\}$ can be considered ϕ-mixing if X_i and X_{i+j} become virtually independent as j becomes large. [14].

After collecting a fixed number of steady-state observations and forming 96 batches, the batch means are computed. The first two batches are discarded, an the independence of the remaining batch means is tested. If the test fails, the batch means are tested for joint normality. If this second test fails, the batch size is increased and the procedure is repeated until one of the tests succeeds. Upon acceptance of the joint multivariate normality, certainly the batch size is not the optimal one, so a correction to adjust the classical batch means confidence interval should be applied. The authors used an inverted Cornish-Fisher expansion of the NOBM t-statistic to accomplish this task.

Chen and Kelton [19] proposed a heuristic sequential procedure for controlling the run length of stochastic simulation so that the mean estimate satisfies a pre-specified precision requirement. The batching is based on systematic sampling and ϕ-mixing conditions. The procedure makes use of the runs-up test proposed by Knuth [78] for testing whether the means appear to be independent. The empirical results of the authors showed reasonable coverage, though they recognized the variance of the simulation run length is large, due the lag length is doubled at each interaction. even a 90% confidence estimator.

Yeh and Schmeiser [133] developed a sequential procedure of nonoverlapping batch means denoted as Dynamic Batch Means (DBM). Instead of keeping each output observation, DBM stores the sum of each batch in a vector of size $2k$ cells, specified by the user. Always when the vector is full, DBM collapses the batch sums into k cells, and the batch size is updated. At any given time, by means of the actual sample size n and k, DBM determines the full batch size m, the current batch c, and the number p of observations in the current batch. Estimation of the variance of the sample uses only the b full batches by applying the correction factor $(bm)/n$.

The replicated batch means (RBM) was presented by Andradóttis [6]. By collecting n observations in $r > 1$ independent replications, RBM forms in b batches of size m in each replication, that is, RBM produces rb batches that are used to yield a point and an interval estimator. The author claimed that in case of simulating a stochastic process with positive serial correlation, RBM offers a smaller variance than the classical point estimator of NOBM. RBM also has a better coverage than NOBM, at least for the EAR(1) process at a 90% confidence level.

Yeh and Schmeiser [134] compared NOBM, OBM (complete and partial), and STS-area estimators based on batching, using a single processor environment, and concluded that OBM outcomes the other ones in the sense it yields small MSE for the

estimator of the variance of the sample mean, while in [95] the reader can find a perspective of batching methods in environments of **MRIP**.

2.5 Parallel simulation

Parallel processing has been and will continue to be a dominant feature of modern computer architectures. To achieve the intended goal of speeding up processing tasks, one needs to alleviate issues related with communication and synchronization. Concerning quantitative stochastic simulation, the problem of interest is to determine the amount of parallelism that can be effectively exploited, that is, to find a suitable combination of factors so that a large number of processors can be used to execute the simulation efficiently while producing an accurate estimate of a system performance parameter θ.

Haynes et al. [64] identified simulation as a promising area for parallel processing, since the computation for generating each observation X_i (used to estimate μ), are repeated over and over. Bhavsar and Isaac [13] pointed out that, in order to exploit this intrinsic parallelism, one could assign the computations among computers in parallel.

Heidelberger [66] investigated statistical issues that arise when discrete event simulations are run on parallel processing computers. By running parallel independent replications, he obtained some estimators by applying the following stopping rules :

- Complete N replications.

  As soon as a processor finishes producing an estimate, it starts a new replication. The first N completed replications are used for the estimation. There is a possibility of sampling bias, when the measures are not independent of the run time. Consider estimating the average of the first m waiting times at, say queue 1, in a network of queues with probabilistic routing and feedback. For example, in one run it may take 100 arrivals to get m waiting times at queue 1, but in another run, it may take only 50 arrivals - the run requiring only 50 arrivals would complete earlier, so with the sampling rule we would tend to get more samples associated with runs requiring fewer arrivals. This would lead to the bias, since for example, if there are fewer arrivals, the queues might be shorter and thus the waiting times shorter than are typical. On the other hand, if the output of the simulation is statistically independent of the run time (e.g. the average of the first m waiting times of a single FCFS queue), then there would be no sampling bias.

- Start N replications.

  N replications are started and the procedure waits their completions without starting new replications, i.e. only those N replications are used for the estimation. Moreover, replications are ordered by their starting times rather by their completion times. Although this stopping rule has better statistical properties than the previous one, the price to be paid for using it is an increased run length.

- Fixed number of replications per processor

  Also known as *static completion assignment*. Let P be the number of processors and let X_{ij} be the observation from the jth replication that runs on processor i. Each processor is assigned a fixed number R of replications, therefore, the total number of replications is n=RxP. Because $X'_{i,j}s$ are IID, the mean estimator

$$\bar{X}_n(P) = \frac{1}{n}\sum_{i=1}^{P}\sum_{j=1}^{R} X_{ij} \tag{2.31}$$

  is both a strongly consistent estimator for $\mathrm{E}[X_{i,j}]$ and asymptotically distributed with mean $\mathrm{E}[X_{i,j}]$ and standard deviation $\sigma(\bar{X})/\sqrt{n}$ as either $m \to \infty$ or $P \to \infty$.

  Generally speaking, assignment could be done in two different manners : (i) all processors cooperating on a single replication of the simulation; (ii) each processor carrying out an independent replication of the same simulation and the results being averaged together. The policies discussed above are related with the second manner of assignment.

  The former is known as Single Replication in Parallel (SRIP) [105], and the rationale is to break up the model itself across the processors. Synchronization between the submodels seems to be the great challenge of this approach, and the structure of some models inherently limits the attainable simulation speedup [128]. SRIP is out of the scope of this work, but excellent introduction can be found in [92] [46] [47].

  The later is known as Multiple Replications in Parallel (**MRIP**), and was proposed by Pawlikowski et al. [106] for primarily studying performance of telecommunication networks.

  MRIP is conceptually simple, relative straightforward to implement and statistically efficient. A very important feature of the **MRIP** approach is its natural solution regarding fault tolerance, since a loss of one or more processors does not crash the whole simulation experiment, provided that at least one processor remains able to continue submitting data to the global analyzer [106].

  A possible drawback would be the necessity of sufficient memory to hold the entire simulation [46], but that seems not to be so critical nowadays. In the next subsection we describe briefly a software package that implements an **MRIP** scenario that will be used as a basis to our experimental investigation.

  Heidelberger [65] developed a framework to compare the statistical efficiency of these two approaches for estimating steady-state quantities. This analysis showed, qualitatively, that parallel replications approach is statistically more efficient than distributed simulation, provided that initialization bias are conveniently treated.

2.5.1 An MRIP Implementation

We are going to argue taking into account Akaroa-2, an **MRIP** implementation designed at the Department of Computer Science of the University of Canterbury, in

Christchurch, New Zealand, for full automatic parallelization of common sequential simulation models, and full automated control of run length for accuracy of the final results [33].

An instance of a sequential simulation model is launched on a number of workstations (operating as simulation engines) connected via a network, and a central process takes care of collecting asynchronously intermediate estimates from each processor and calculates conveniently an overall estimate.

The only things synchronized in Akaroa-2 are substreams of pseudorandom numbers to avoid overlapping among them, and the load of the same simulation model into the memory of different processors, but in general this time can be considered negligible and imposes no obstacle.

Akaroa-2 enables the same simulation model be executed in different processors in parallel, aiming to produce IID observations by initiating each replication with strictly nonoverlapping streams of pseudorandom numbers provided by a multiplicative congruential generator with modulus of $M = 2^{31} - 1$, exhaustive tested by Fishman and Moore [41]. The latest version of Akaroa-2 uses longer than sequences of pseudorandom numbers.

Essentially, a master process (*Akmaster*) is started on a processor, which acts as a manager, while one or more slave processes (*akslave*) are started on each processor that takes part in the simulation experiment, forming a pool of simulation engines (see Figure 1). Akaroa-2 takes care of the fundamental tasks of launching the same simulation model on the processors belonging to that pool, controlling the whole experiment and offering an automated control of the accuracy of the simulation output.

At the beginning, a stationary test (due to Schruben [121]) is applied locally within each replication, to determine the onset of steady state conditions in each time-stream separately and the sequential version of a **CIP** is used to estimate the variance of local estimators at consecutive checkpoints, each simulation engine following its own sequence of checkpoints.

Each simulation engine keeps on generating output observations, and when the amount of collected observations is sufficient to yield a reasonable estimate, we say that a checkpoint is achieved, and it is time the local analyzer to submit an estimate to the global analyzer, located in the processor running akmaster.

The global analyzer calculates a global estimate, based on local estimates delivered by individual engines, and verifies if the required precision was reached, in which case the overall simulation is finished. Otherwise, more local observations are required, so simulation engines continue their activities.

Whenever a checkpoint is achieved, the current local estimate and its variance are sent to the global analyzer that computes the current value of the global estimate and its precision. A checkpoint is associated with an estimate of type

$$(N_i, X_i, V_i),$$

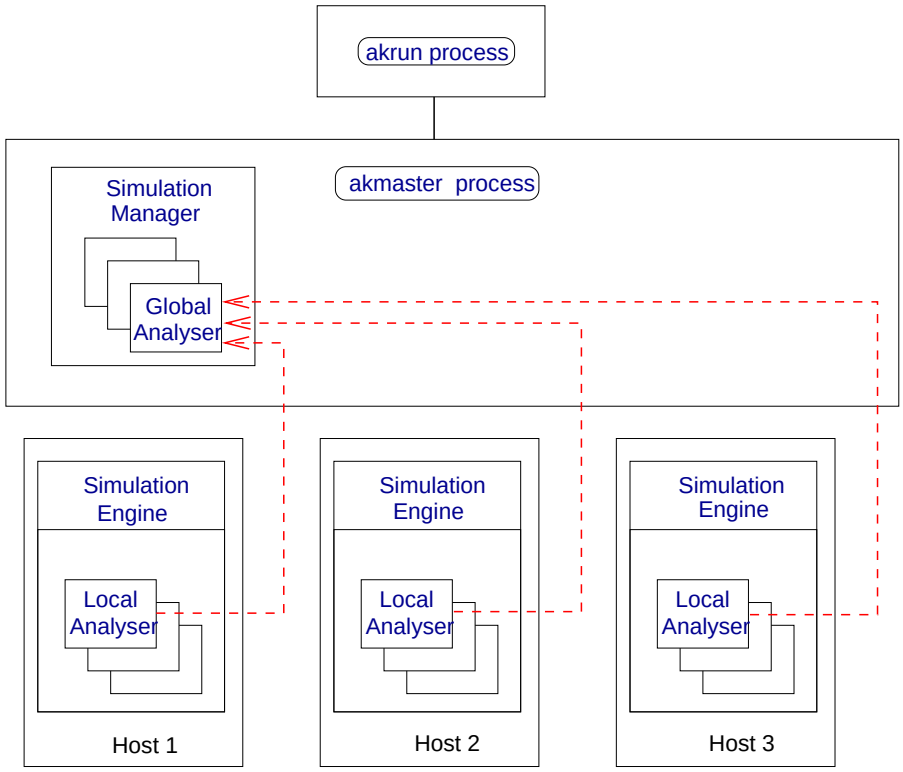


Fig. 1: Schematic diagram of Akaroa

where N_i is the number of observations collected by processor i until that moment (minus the observations discarded in the transient phase), X_i is the sample mean and V_i is the variance of the sample mean, obtained by applying one of the methods of analysis mentioned above. Particularly, we are interested in methods based on batching approach, as they are widely accepted, but feasibility of its application on a parallel environment (as that implemented by Akaroa-2) requires deeper investigation.

There is a global analyzer (for each performance parameter being estimated) that averages the estimates coming from the simulation engines. A natural manner of averaging adequately several estimates coming from several simulation engines is to form a linear combination of the estimates. The choice of weights can produce unbiased estimator of the global mean and its variance. One of this choice, implemented and commonly used in Akaroa-2 is

$$\bar{\bar{X}} = \frac{\sum N_i.X_i}{\sum N_i} \tag{2.32}$$

$$Var\{\bar{\bar{X}}\} = \frac{\sum N_i^2.Vi}{\sum N_i^2} \tag{2.33}$$

2.5.2 Performance issues

It is useful to know which factors yield direct influence on the efficiency of a parallel computing environment. Quality of hardware is a very subjective criterion, but communication among processes, on the other hand, is of extreme relevance. As this was one major issue underlying the design of Akaroa, we can take it for granted.

Our considerations are restricted to steady-state estimation problem. Specifically, we consider sampling schemes that delete some initial part of each run in order to reduce *initialization bias* since the model cannot typically be started in its steady-state distribution.

Ewing et al. [32] showed some potential limitations of classical batch-means-based **CIP** under **MRIP**. The challenge is to find a computationally not too intensive method for selecting the batch size, based on a trade-off between accuracy of the batch size and computational complexity of the procedure.

We shall see that OBM and CSUM can be tuned in such a way to compensate the negative effects of not complete fulfillment of the underlying assumptions (perfect independence among batch means in the former, and normality among the random variables A_i, in the latter), characterizing them as robust methods of analysis. This is feasible due two new factors that enter in scene :

1. the degree of parallelization;
2. the possibility of improving batching estimators by increasing the degree of overlapping (for OBM only).

The benefits and cost of this factors have to be carefully analyzed and quantified.

Intuitively, when using the method of parallel replications on a large number of processors, one expects to get final estimates after only a relatively short amount of time. However, there are still a number of statistical problems associated with estimation of steady-state parameter of stochastic processes through stochastic simulation using **MRIP**. These problems are, fundamentally, the same found in a single-processor simulation environment, but they are augmented due parallelism. The problems basically arise because any bias effects can be magnified on highly parallel machines. Refer to works of Heidelberger and Welch [66] for a theoretical analysis of this issue.

3

Output Analysis Methodology

3.1 Introduction

Chapter 2 reviewed the literature necessary to complete understanding of this thesis. Chapter 3 describes the combined approaches used in this research to give orientation for proposing, designing and evaluating the performance and feasibility of some sequential confidence interval procedures that can be run under MRIP.

A brief overview of the methodology was provided in section 1.5; this chapter aims to build on that introduction and to provide details of each decision adopted in the design and followed in the experimental phase of this research. The chapter is organized around the combination of three major approaches : the framework proposed by Schriber and Andrews [116], the sequential coverage analysis proposed by Pawlikowski, McNickle and Ewing [104], and the coverage function proposed by Schruben [117]. Together they formed the cornerstone methodology followed in this research.

3.2 A conceptual framework

Reporting the effectiveness of methodologies for confidence interval procedures, as well as testing their validity was frequently confusing and not always comprehensive and comparable. To structure the methods of constructing confidence intervals, Schriber and Andrews [116] proposed a conceptual framework, which will be used throughout this thesis. Figure 2 shows a schematic representation of this methodology that we are going to summarize in the sequence. Dashed figures stands for not yet implemented facility.

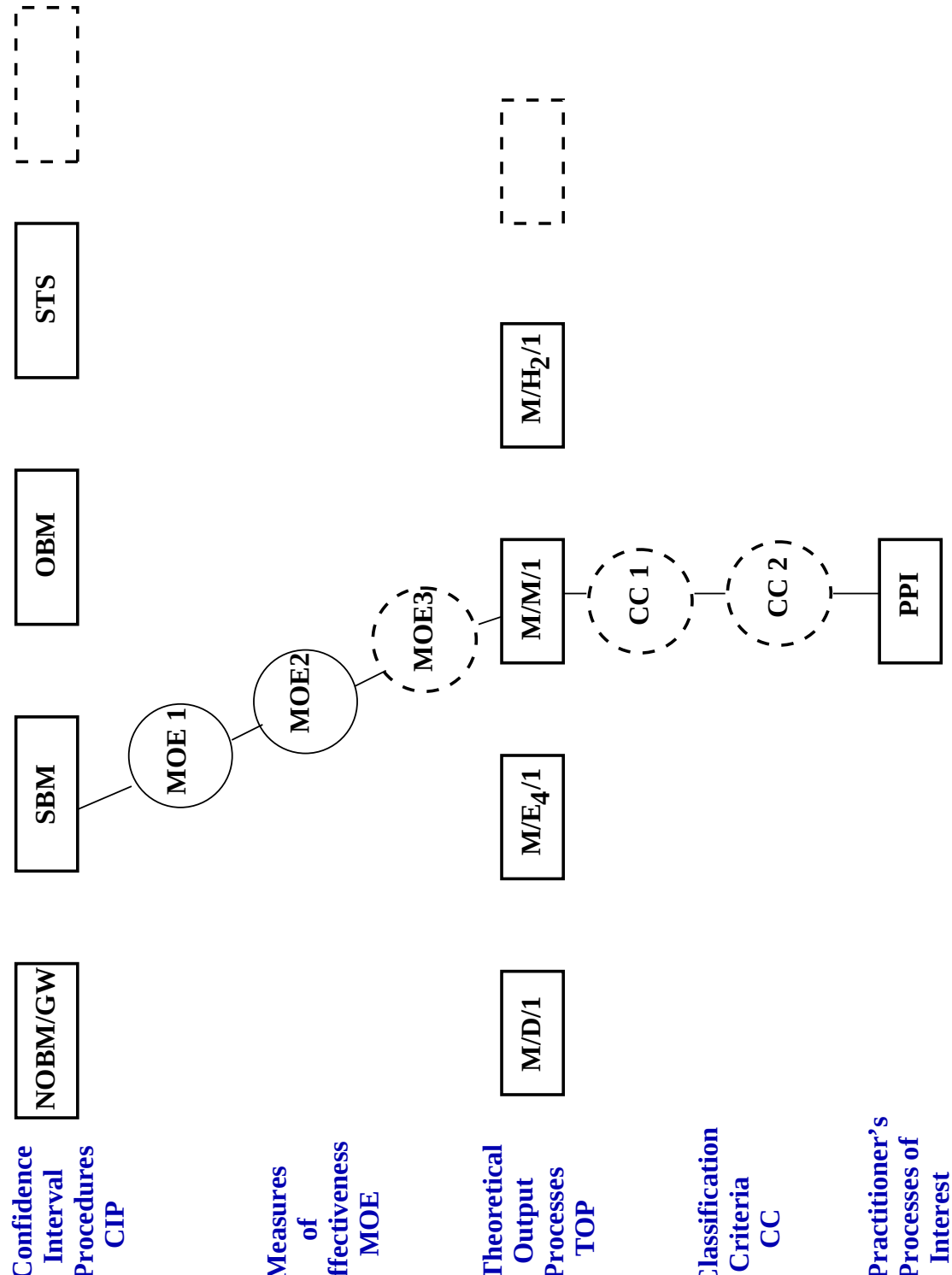


Fig. 2: The conceptual framework proposed by Schriber and Andrews [116]

CIP The first row of the framework lists the confidence interval procedures (CIPs) being investigated. The implemented ones were explained in section (2.3).

MOE To evaluate the behavior of each CIP (and make it possible the comparison of their performance under MRIP), we must choose a set of measures of effectiveness (MOE). For the sake of clarity, figure (2) shows one of such set of measures for a single CIP.

The main measure of effectiveness of a CIP is the coverage. Coverage rates tend to be substantially lower than the desired confidence level, even in those cases in which procedures have an asymptotically consistent large sample theory [51]. Usually, empirical studies of the efficiency of a new CIP measure the coverage of a known parameter at a particular confidence level, e.g. 90%. Schruben [117] demonstrated that some undesirable effects may occur for other values of confidence level and, then, suggested the construction of an empirical coverage function for a wider range of confidence level.

The second MOE of our interest is the sample size required by a CIP to give a point and interval estimate with a desired precision. The smaller sample size, the more efficient is the procedure, provided the coverage is acceptable.

The third main MOE is the variability of the length of the final confidence intervals. For all simulation experiments we have adopted a 5%-relative precision as a stopping rule, defined as $H/\bar{X}(n)$. Thus, expected value of the confidence interval half-width, which measures the accuracy of the estimation, is not an issue in our experiments.

One can assess empirically this MOE by means of the coefficient of variation, defined as the sample standard deviation expressed as a percentage of the sample mean as shown in

$$CoV\{H\} = \frac{\sqrt{Var\{H\}}}{E\{H\}} \tag{3.1}$$

As we shall see later, MRIP is a natural variance reductor, thus it seems natural to assess the speedup as a fourth important MOE. A set of experiments that emphasize the effect of applying a different number of processors will be investigated.

TOP For analyzing coverage we need a stochastic process with known analytical solution (TOP – theoretical output process). We have chosen queuing systems with increasing coefficient of variation of the service times C_x (see Fig. 3). In Appendix C, the reader finds some details of the queuing models we adopted.

Larger C_x results in a more correlated process generated, situation where most CIPs find difficulties.

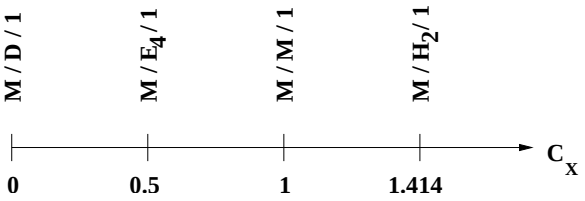


Fig. 3: Queuing models with increasing coefficient of variation of service times

M/D/1 : the arrivals are Poisson and the service time is a fixed, deterministic quantity. This is actually a reasonable model for the processing of fixed length packets in a communication network, such as ATM.

M/E$_4$/1 : the arrivals are Poisson and the service facility consists of four serial, identical, exponentially distributed service stages. In many communication systems, packets do not arrive individually, rather they still come in a Poisson mode but in fixed-size bulk arrivals. Only one packet will be in the service at any time. We can model this kind of systems by means of $M/E_k/1$, where k is the bulk size [77].

M/M/1 : probably the most commonly used queue. It is assumed that the interarrival times and the service are exponentially distributed. Suppose there are m independent Poisson data streams, each supplying packets at rate λ/m, arriving at a common "concentrator" where they are mixed into a single data stream of combined rate λ. Suppose that packet lengths are independent and exponentially distributed so that packet transmission times are exponentially distributed with mean transmission time, say, $1/\mu$. The concentrator then forms an M/M/1 system which statistically multiplexes the independent data stream into a single data stream.

M/H$_2$/1 : the arrivals are Poisson and the service facility consists of two parallel service stages. This is a reasonable model for representing systems that may service more than one packet at a time, such as wavelength division multiplexing.

PPI In the last row we find the practitioner's processes of interest, i.e., the real world problems we are ultimately interested in applying the investigated methods of analysis. In this research, the PPI is an adaptation of the "Simultaneous MAC Packet Transmission (SMPT)" approach to a CDMA based mobile communication system with a varying number of mobile users. This problem was proposed by Fitzek and Wolisz [42].

CC In order to choose which TOP behaves more closely to a given PPI, we need some criteria of classification. Schriber and Andrews suggest the construction

of three functions : the autocorrelation funtion, the partial autocorrelation function and the spectral density function. These functions should be "*compared to the corresponding functions for different classes of TOP's until a similarity is subjectively discovered*". Then, the supposed similarity should be objectively examined by using a chi-square test of goodness of fit.

We have adopted practical rules of thumb based on a vast amount of repetitions of simulations of the selected queuing systems. We observed that transient phase is usually not so long in our PPI. Moreover, it takes very long time to get an accurate estimate (e.g. when the confidence level is 0.95, and traffic intensity is high). This effect can be seen for example when simulating $M/H_2/1$ queueing system. This way, CIPs with best performance for this scenario will be used in the case study of Chapter 6.

3.3 Coverage analysis

Coverage of confidence intervals is defined as the relative frequency with which the final confidence interval contains the true value. Experimental analysis of coverage is essential for assessing the quality of practical implementations of methods used for determining confidence intervals in steady-state simulation. A confidence interval for the coverage can be estimated by means of [85] [104] :

$$\left(c - z_{1-\alpha/2}\sqrt{\frac{c(1-c)}{n_c}}, c + z_{1-\alpha/2}\sqrt{\frac{c(1-c)}{n_c}} \right) \tag{3.2}$$

where c is the coverage, $z_{1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of the standard normal distribution and n_c is the number of repetitions of the coverage experiments.

Traditionally, coverage has been done in a fixed-sample basis. It is evident that many information can be lost and coverage should be analyzed in a sequential manner. Pawlikowski et al. [104] proposed a sequential coverage analysis based upon three rules :

$\mathbf{R_1}$ Analysis of coverage should be stopped when the relative precision of the estimated coverage satisfies a predetermined level. The same arguments used for the sequential analysis of the mean of the output process can be used here : One can control the statistical error of the analysis.

$\mathbf{R_2}$ To be representative, the analysis should start only after collecting a minimum number of "bad" confidence intervals. It is a safe way the authors found to guarantee that the sample used for the coverage analysis is statistically acceptable. All experiments in this research adopted a minimum number of "bad" confidence intervals equal 200.

R_3 Results from too short runs should be discarded and not taken into account, since, by chance, the stopping rule can be achieved too early, which can produce bias in the coverage estimate.

Refer to [93] [94] [95] [104] for some examples of successful applicability of this approach.

3.4 Coverage function

Confining our discussion to the one-parameter case, intuitively we want an estimator to yield estimates whose distribution is close in some sense to the real parameter. This broad criterion is not sufficient to resolve ambiguities.

CIPs are based on some assumptions in order to generate valid confidence intervals, and deviations from these assumptions might affect their validity. It is important to investigate the correctness of the assumptions taken by a CIP being studied, by means of a detailed coverage analysis. Since we are dealing with various approximations, we are investigating how well a CIP behaves for a range of confidence levels η, i.e. how robust it can be.

When the assumptions are satisfied, the coverage η has a uniform distribution. Schruben [117] suggests the construction of an empirical distribution function $G_{\eta^*}(\eta)$ with the observed values of confidence level η^* for a sample size n. This empirical function carries more information than a single value of coverage, e.g. at $\eta = 0.9$, a common practice to summarize empirical results in coverage analysis. Ideally, $G_{\eta^*}(\eta) = \eta$, but usually two other situations can arise :

- $G_{\eta^*}(\eta) < \eta$, when positive serial correlation has been ignored, which means that the final η was overestimated.
- $G_{\eta^*}(\eta) > \eta$, when negative serial correlation has been ignored. This can lead to waste of time in sequential procedures, since it may be connected with collecting more observations than necessary for achieving a desired precision.

Let D be the relative deviation of the observed confidence level from the theoretical value, defined as $D = \frac{D^+ + D^-}{\mu}$, where

$$D^+ = max(G_{\eta^*}(\eta) - \eta, 0) \quad \text{and}$$

$$D^- = max(\eta - G_{\eta^*}(\eta), 0)$$

Large values of D^+ suggest that the procedure generates confidence intervals larger than necessary and, therefore, the procedure is not efficient. Large values of D^- suggest that the coverage frequency is lower than intended and the intervals are

not valid. Since $D^+ \geq 0$ and $D^- \geq 0$, if deviations above and below the nominal coverage are quite small, $D \rightarrow 0$ and the coverage estimate is accurate. Any reasonable CIP should have perfect coverage if the confidence level is set high enough. By perfect coverage we mean the observed coverage approaches the nominal coverage.

3.5 Statistical techniques

Statistical techniques are extremely useful in system performance evaluation. Confidence interval procedures make use of some kind of statistical technique to certify their hypotheses. A simple substitution of an statistical technique to another one is enough to consider the resulting CIP a different procedure, with probably different statistical performance. Among the tasks that such a statistical technique can deal with inside a CIP, we can refer to briefly :

1. the end of the transient phase;
2. the independence among simulation output data;
3. how long a simulation run should be.

The selected techniques to solve this issues in this research are discussed below.

3.5.1 Stationarity test

The challenge in the first part of each CIP is to determine the truncation point beyond which the remaining sequence preserves much of the original one and is consistent with the objective of minimizing bias.

Gafarian et al. [48] surveyed the practical rules (available at that time) for determining the end of the transient period, and found that none of them performed well in the practice. Stacey et al. [124] share this conclusion when they claim that "*these techniques behave quite badly when applied in practical simulation experiments, especially when simulating heavily loaded systems*".

Lee and Oh [88] investigated two methods based on the Chaos Theory. Simulating the average waiting time of an M/M/1 queuing system, they concluded that the truncation point cannot be found in some cases.

A number of formal initialization bias tests have been developed that try to determine if a process contains an initial transient [99] [98]. These initial transient bias test are typically hypothesis tests with null hypothesis, H_o : no initialization bias presents, and alternative hypothesis, H_1 : initialization bias present.

One of the most successful formal methods for estimating the length of the initial transient period, based on statistical tests developed for testing stationarity of time series proposed by Schruben et al. [121] can be found implemented in Akaroa-2,

and we applied it in our experimental investigation. This statistical procedure tests for remaining bias in simulation output after discarding some initial observations. The test for stationarity performs the following way.

Let n_o be the smallest truncation point we are looking for, for which a first approximation can be found by using one of the heuristic rules presented in [99] So stated, the test verifies whether the remaining n_t observations (after n_o) can be considered a sample from a covariance stationary process.

A tactical problem for this test arises because we should know the variance $\sigma^2[\bar{X}(n)]$ and the number *df* of degrees of freedom for its χ^2 distribution. Pawlikowski [99] suggested that a robust estimate of the variance can be obtained if the estimation use only a subsequence of the last n_v of the n_t observations.

If the test fails, the collected observations are discarded, and another value of n_o is chosen. The stationarity portion test is repeated. Practical values to these quantities and the pseudocode implemented in Akaroa-2 is given in [99].

3.5.2 Jackknife method

Suppose that an estimator $\hat{\nu}$ can be computed from m observations x_j (j=1,...,m). Jackknife method groups those m observations into N groups of equal size M (N may be equal to m, so M=1). Then, it eliminates one group of observations and calculates the same estimator, say $\hat{\nu_i}$ (i=1,...,N), from the remaining (N - 1)M observations.

The *pseudovalue* J_i is defined as the following linear combination :

$$J_i = N\hat{\nu} - (N-1)\hat{\nu_i} \tag{3.3}$$

The jackknife estimator is defined as the average pseudovalue :

$$\bar{J} = \frac{1}{N}\sum_{i=1}^{N} J_i = N\hat{\nu} - (N-1)\sum_{i=1}^{N}\frac{\hat{\nu_i}}{N} \tag{3.4}$$

It can be proved that if $\hat{\nu}$ is biased, the jackknife estimator $\bar{J}$ is less biased [75].

3.5.3 Von Neumann's test

Let $\{\bar{X_i}(m) : i = 1, 2, \ldots, b\}$ be a sequence of b batch means of size m for a certain sample size n. To test the level of correlation among the batch means, we compute the following statistic :

$$C_b = 1 - \frac{\sum_{i=1}^{b-1}(\bar{X}_i(m) - \bar{X}_{i+1}(m))^2}{2\sum_{i=1}^{b}(\bar{X}_i(m) - \bar{X}(n))^2}$$

If $\bar{X}i(m)$ are normal, then under null hypothesis H_o of independence, C_b has mean zero and variance $(b-2)/(b^2-1)$. For a specified quantile $(1-\beta)$ from the standard normal distribution, if $C_b > z_\beta\sqrt{(b-2)/(b^2-1)}$, then the null hypothesis H_o can not be accepted. The batch size should be increased, more observations should be collected until completing the new sample size, and the von Neumann's test should be repeated.

F_{C_b}, the distribution of C_b, should become already close to normal for small number of batches (say $b \geq 8$). Kleijnen et al. [76] emphasized that the power of this test statistic is small if the number of batches is less than 100.

Hypothesis concerning variance can give poor results if population distribution deviates appreciably from the normal distribution [126]. Therefore, one can make use of a t-test for the mean, which also uses an estimate of the variance.

Since rank procedures are less computational demanding, an interesting idea is to apply von Neumann's test to rank-transformed data, the so-called Bartels test [11]. If R_i is the rank of the observation X_i, then the rank version of von Neumann's test is given by

$$RVN = \frac{\sum_{i=1}^{n-1}(R_1 - R_{i+1})^2}{\sum_{i=1}^{n}(R_i - \bar{R})^2} \tag{3.5}$$

Appendix B gives the critical values for the Rank version of von Neumann's test. Bartels showed that the rank version of von Neumann's test is more powerful compared to the nonrank version when the distributional assumptions of normality are not satisfied. Very little is lost when the distributional assumptions for normality are fully satisfied.

3.5.4 Test for normality

Tests for normality are very sensitive to sample size so can easily be misleading (e.g. rejecting normality for minor deviations in large samples or failing to reject for major deviations in small samples). The best test, in the sense of being powerful against a broad range of alternative distributions, is the Shapiro-Wilk test [122].

Given a sequence X_i (i=1,... n) :

1. Sort the sequence in increasing order, yielding $Y_1 \leq Y_2 \leq \ldots \leq Y_n$.
2. Compute

$$S^2 = \sum_{i=1}^{n}(Y_i - \bar{Y})^2 \tag{3.6}$$

3. If n is even, n = 2k (if n is odd, n = 2k +1) compute

$$b = \sum_{i=1}^{\lceil n/2 \rceil}(Y_{n-i+1} - Y_i)a_{in} \tag{3.7}$$

where the coefficients a_{in} are given in Appendix A.

4. Compute

$$W = \frac{b^2}{S^2} \tag{3.8}$$

5. Quantiles of the distribution of W is given in Appendix A.

   Small values of W are significant, i.e., indicate non-normality and the hypothesis that the respective distribution is normal should be rejected.

3.6 Applicability of these statistical techniques

All CIPs here investigated take advantage of the Schuruben's stationarity test to determine when the transient period finished and, consequently the steady tate phase begins. For application of this test, we did not make any attempt to tune it, that is, no parameter was changed in order to improve the results (if it could at all). We assume that the configuration and implementation proposed in [99] is suitable for the kind of queueing models we have investigated.

Jackknife methodology was applied in the sequential CIPs based on Batch Means to estimate the correlation coefficients. Alternative to that nonparametric method could be the bootstrap, but this last one is much more time-consuming. There is only one case when jackknife fails [31], namely, when the statistic to be estimated is not smooth[1], but it is not the case of correlation coefficients.

Jackknife estimators, though time-consuming, yields less biased estimates, but it requires at least 100 batches to perform suitably. A parametric alternative we

[1]small changes in the data set cause only small change in the statistic

have selected was the test of independence based on the statistic of von Neumann. We found it interesting because its rank version can work with small number of batches as short as 8, for example. It means that while Type II error is greater than zero, the procedure can leave quicker the phase of determining the batch size, and enter the estimation phase where the simulation engines work effectively in parallel. There is a possibility to compensate the imprecise batch size selection due a small number of batches, by increasing the degree of parallelization. In fact, we have proved in Section 5.9 that this idea is correct.

Methods based on STS also need to determine the batch in such a way that the asymptotic random variables A_i's associated with each batch can be considered normally distributed. Studies such as the Monte Carlo study of Shapiro and others [123] have consistently shown that for testing goodness-of-fit of normal distributions, the Shapiro-Wilk statistic has superior power to other statistics in detecting that the data comes from a wide range of other distributions.

4

Design of sequential batching procedures

4.1 Introduction

Chapter 3 listed well established techniques and statistical tests that support the methodology we are going to apply for validating and assessing the performance of the sequential procedures we propose in this research. In this chapter, we look over the the implementation details, and the relevant design decisions as well. As stated in [59], care must be taken in the design and implementation of sequential procedures to avoid inappropriate early termination. Sequential procedures, on the other hand, tend to become asymptotically valid when the run lengths are relatively long, since the width of the confidence interval approaches zero.

In sequential procedures the run length of simulation is a random variable itself, as it depends on the output observations, and one has no longer direct control of the amount of simulation time. The procedures we are going to propose in this chapter rely on the assumption that the output process is stationary in both mean and variance, since the presence of an initial transient response leads to both biased estimate of steady state mean and to decreased coverage of the estimated confidence intervals.

Although the negative effects of initialization bias are mitigated as the simulation run length increases, the convergence may be slow [23], therefore the procedures apply a test for detecting the extension of the transient period. The procedures investigated by this research can be divided into two parts :

1. Sequential determination of the length of transient period (also known as output process truncation), by means of a stationarity test proposed by Schruben et al. [121] which explores the application of Brownian bridge statistics to the initial transient problem. Observations of the transient period are then discarded.

2. Sequential determination of the length of the simulation run that depends on the desired accuracy requested by the user. We assume here that a single parameter is estimated, namely the mean average time spent by a customer in a queuing system.

As a first approximation to the truncation point n_o, we have applied the following heuristic rule proposed by Fishman [37] :

> *The initial transient is over after after n_o observations if the time series $x_1, x_2, \ldots, x_{no}$ crosses the mean $\bar{X}(n_o)$ k times.*

By simulating some queuing systems for assessing the quality of our procedures, we followed the recommendation of Gafarian et al.[1] [48], applied successfully by Pawlikowski and Asgarkhani [9] [101], and set $k := 25$.

4.2 Sequential procedures based on NOBM

Periodically, a sequential procedure collects estimates to verify whether the stopping rule has been achieved or, in terms of MRIP, whether sufficient observations have been collected in order to yield a reasonable intermediate estimate. At this moment, n observations are available, distributed in b contiguous, nonoverlapped batches of size m ($n = m.b$).

An important parameter in batching-based techniques is the selection of batch size. The choice of the batch size has a direct impact on the quality of variance estimator and the confidence interval [114]. Waiting times from a queue with a higher traffic intensity are more positively correlated; in general, a larger batch size is required if data are more correlated [21].

In order to select the optimal batch size m^* which yields almost uncorrelated batch means at a specified significance level β ($0 < \beta < 1$), correlation coefficients $\hat{\rho}_k$ of all possible lags k (k = 1, 2, ..., L) could be estimated and compared against β. The null hypothesis of statistically negligible correlation among batch means is accepted if [1] :

$$|\hat{\rho}_k| \quad < \quad z_{1-\beta_k/2}\sqrt{v\hat{a}r(\hat{\rho}_k)} \tag{4.1}$$

[1] for an $M/M/1/\infty$ queuing system

where

β_k is the significance level of each of the individual tests ($\beta_k := \beta/L$),

$z_{1-\beta_k/2}$ is the upper $(1-\beta_k/2)$ critical point of the standard normal distribution,

$\hat{\rho}_k$ is the sample kth lag autocorrelation given by

$$\hat{\rho}_k := \frac{1}{b}\sum_{i=1}^{b-k}\frac{(X_i - \bar{X}(n))(X_{i+k} - \bar{X}(n))}{S^2(b)} \tag{4.2}$$

where $S^2(b)$ is the sample variance of the batch means, given in Equation 2.10, and

$\hat{var}(\hat{\rho}_k)$ is the variance of the sample kth lag autocorrelation, estimated by Bartlett's approximation

$$\hat{var}(\hat{\rho}_k) := \frac{1}{b}\left\{1 + 2\sum_{v=1}^{k-1}\rho_v^2\right\} \tag{4.3}$$

Obviously, it would make such test not feasible, so one can estimate L autocorrelation coefficients of lag k ($k = 1, 2, \ldots, L$) by means of jackknife estimators, which are usually less biased than the ordinary autocorrelation coefficient estimators [91]. L should not exceed 10% of the number of batches [49].

By applying jackknife estimators as a secure method of improving the quality of correlation coefficients estimation, we come across the limitation imposed by the number of batches it should touch with. If this number is very large, nonparametric calculations can be prohibitive. On the other hand, keeping the same number of lags can degrade the decision about independence.

To get acceptable estimators of the correlation coefficients, at least 50 batches should be available [16]; thus in the case of jackknife estimators one should use at least 100 batches [99].

4.2.1 Classical NOBM

Pseudocode 1 presents the sequential run length control procedure based on NOBM, already implemented in Akaroa-2. Fig. 4 is the corresponding flowchart. The variables are explained in Tab. 1 :

The procedure collects n steady-state observations, and divide them into b contiguous nonoverlapping batches of size m. For the sake of simplicity in terms of implementation, $n = b.m$.

m	initial batch size indicated by the user;
b	number of batches indicated by the user; due the jackknife estimators, this number must be at least 100;
$\boldsymbol{\epsilon}$	the relative precision requested by the user;
L	number of autocorrelation coefficients;
$\boldsymbol{\beta}$	overall significance level of L tests against correlation;
$\boldsymbol{\beta_k}$	significance level of each of the individual tests against correlation;
contM	counts the observations inside each batch;
contB	counts the number of batches;
n	sample size;
t	grouping factor; batch size will increase as a multiple of the initial batch size;
Uncorrelated	boolean variable indicating whether the batch means can be considered uncorrelated;
threshold	reference value for accepting that correlation among batch means is negligible;
$\boldsymbol{z_{1-\beta_k/2}}$	upper $(1-\beta_k/2)$ critical point of the standard normal distribution.
$\hat{\rho}_k$	correlation coefficient of lag k;
Bmean	accumulates observations inside a batch;
BM	array containing the batch means;

Tab. 1: Variables of the pseudocode for NOBM.

To assure that batch means are approximately independent, the procedure tests the correlation coefficients for lag $k(k = 1, \ldots, L)$ against correlation. If these correlations can not be considered negligible at a significance level β, the procedure increments m and collects more observations until the sample size is $n = b.m$. Once again, it divides the observations into b batches and repeats the correlation test. When this test is successful, the procedure accepts the current batch size as the optimal batch size m^*.

Having determined m^* that guarantees acceptably low correlations among the batch means, the procedure can enhance itself by applying Schmeiser's finding and reorganizes the observations into $10 \leq b \leq 30$ batches, and then it enters in the estimation phase to find the point and interval estimator by applying Equation 2.8 to Equation 2.10.

While $H/\bar{X}(n) < \epsilon$, where ϵ is a pre-specified relative precision given by the simulation practitioner, the experiment continues generating batches of size m^*, reorganizing them into $10 \leq b \leq 30$ batches and testing the stopping rule. Each replication executing NOBM uses a single-pass algorithm to compute the batch means, the sample mean, and the the sample variance. The computational complexity required is O(n) and O(1) storage.

Concerning the test of independence, as we have already said, we estimated the correlation coefficients by applying the nonparametric method of jackknifing. It is a robust approach, though it tends to be computationally intensive, but it has

the side benefit of reducing statistical bias of not unbiased estimators. It is a simple manner to improve the quality of correlation coefficients estimation. The goal behind our implementations was to make them as accurate as possible, and then to reduce the completion time by adding more processors to the simulation experiments.

A plenty of tests of independence in time series can be found in the literature; see [5] and [79] for a review. They have been shown tricky, could cost higher level of demand than jackknife, and, more important, when one performs statistical tests, there is a possibility of two types of errors :

Type I An event is not expected to occur but it does; or
Type II An event is expected to occur but it doesn't.

Nevertheless, during the experimentation (see Section 5.9), having in mind our hypothesis H_2[2], we used the statistical test of von Neuman, proposed by Fishman (see Equation 2.30), to collect evidence to validate H_2.

It does not mean it is possible to make a test more powerful by using more processors. Kleijnen et al. [76] proved that the power of von Neuman's test is small if the number of subruns (batches in our case) is less than 100. Of course, using fewer batches, one would speedup up the test, but Type II error would be higher, which would yield a poor coverage. Chapter demonstrates that if one uses few batches in the von Neuman test but increase the degree of parallelization under MRIP, the coverage can be improved.

[2]imprecision can be compensated by putting more processors to work together

Pseudocode 1 Sequential procedure based on classical NOBM

1: **m** := 100; **b** := 100
2: $\epsilon := 0.05$; $\boldsymbol{\beta} := 0.1$
3: $L := 0.1 * \mathbf{b}$
4: $\beta_k := \beta / L$
5: Uncorrelated := false
6: $n := 0$; $contM := 0$; $contB := 0$; $Bmean := 0$; $t := 1$
Require: Observations are in the stationary phase
7: **while** (**not** Uncorrelated) **do**
8: **repeat**
9: **while** ($contM < t * \mathbf{m}$) **do**
10: Collect an observation X
11: $n := n + 1$
12: $contM := contM + 1$
13: $Bmean := Bmean + X$
{This loop remains until collecting a complete batch}
14: **end while**
15: $BM[contB] := Bmean/contM$
16: $contB := contB + 1$; $contM := 0$
{This loop remains until collecting b batches}
17: **until** ($n = b * t * m$)
{Test batch means for correlation at significance level β}
18: Compute correlation coefficients of lags from 1 to L
19: $threshold := z_{1-\beta_k/2}\sqrt{\hat{var}(\hat{\rho}_k)}$
20: **for** $k = 1$ to L **do**
21: **if** $\hat{\rho}_k > threshold$ **then**
22: $t := t + 1$
23: Join each t***m** observations and find their means
24: contB := (t - 1) * contB / t;
25: Save batch means in the first contB positions of BM[ ]
26: $contM := 0$; $Bmean := 0$; Uncorrelated := false; break
27: **else**
28: Uncorrelated := true
29: **end if**
30: **end for**
{This loop remains until batch means can be considered uncorrelated}
31: **end while**
{For estimation phase, $10 \leq b \leq 30$}
32: $b := 25$
33: Form b batches by joining every four batches
34: $t := 4 * t$
35: Compute the sample mean $\bar{X}(n)$ and the half-width confidence interval **H**
36: **while** ($\mathbf{H}/\bar{X}(n) > \epsilon$) **do**
37: Collect t*m observations
38: Form b batches
39: Compute the batch means
40: Compute the sample mean $\bar{X}(n)$ and the confidence interval half-width **H**
41: **end while**
42: StopSimulation = yes

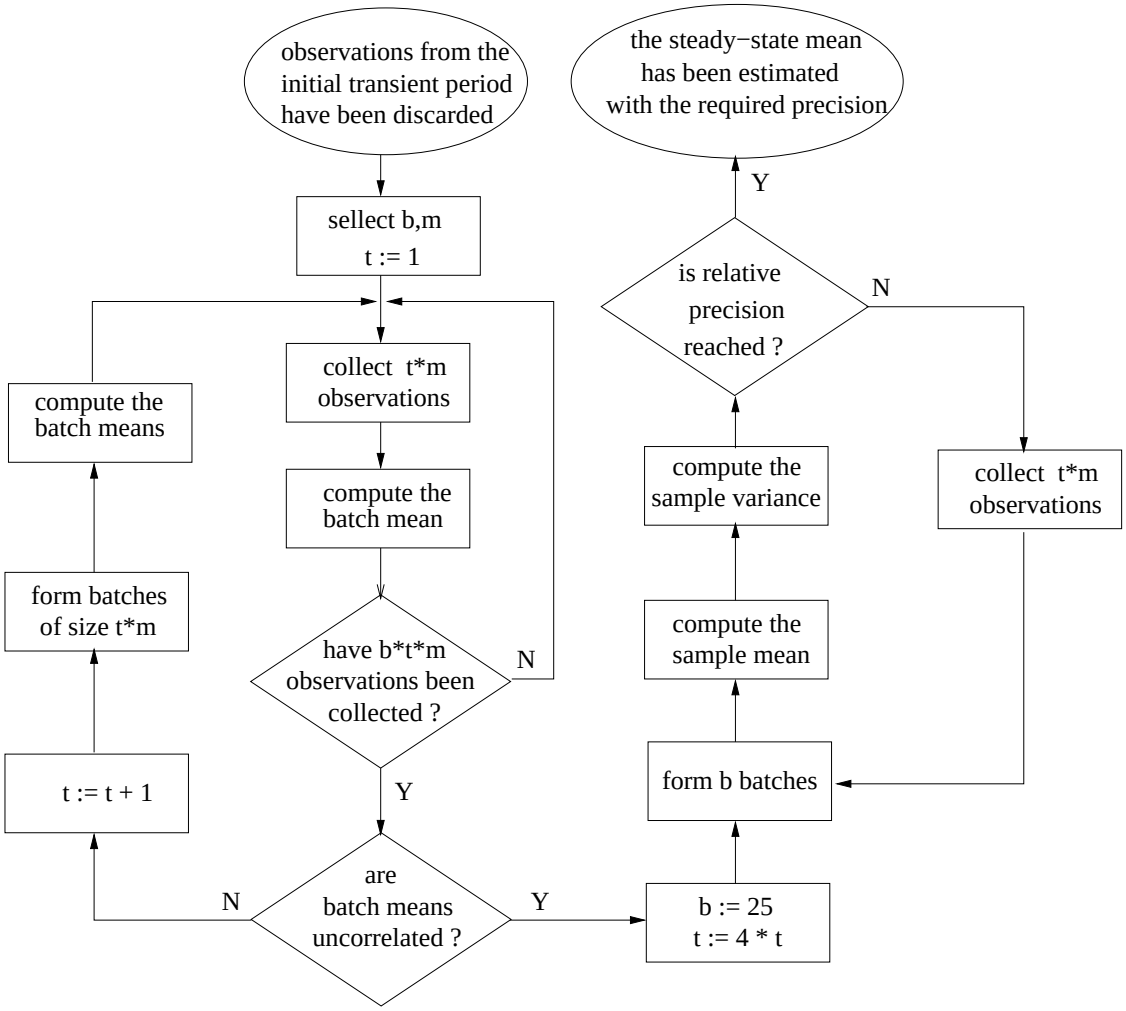


Fig. 4: Flowchart of a sequential procedure based on classical NOBM.

4.2.2 Design of a sequential procedure based on NOBM/GW

Pseudocode 2 presents a proposal of a sequential run length control procedure based on NOBM/GW, and Fig. 6 is the corresponding flowchart. The variables are explained in Tab. 2 :

m	initial batch size indicated by the user;
b	number of batches indicated by the user; due the jackknife estimators, this number must be at least 100;
$\boldsymbol{\epsilon}$	the relative precision requested by the user;
L	number of autocorrelation coefficients;
$\boldsymbol{\beta}$	overall significance level of L tests against correlation;
$\boldsymbol{\beta}_k$	significance level of each of the individual tests against correlation;
contM	counts the observations inside each batch;
contB	counts the number of batches;
n	sample size;
t	grouping factor; batch size will increase as a multiple of the initial batch size;
Uncorrelated	boolean variable indicating whether the batch means can be considered uncorrelated;
Incr	number of additional batches to be added at consecutive checkpoints whenever the stopping rule is not achieved;
threshold	reference value for accepting that correlation among batch means is negligible;
$\boldsymbol{z_{1-\beta_k/2}}$	upper $(1-\beta_k/2)$ critical point of the standard normal distribution.
$\hat{\rho}_k$	correlation coefficient of lag k;
Bmean	accumulates observations inside a batch;
BM	array containing the batch means;

Tab. 2: Variables of the pseudocode for NOBM/GW.

Each replication executing NOBM/GW uses a single-pass algorithm to compute the batch means, the sample mean, and the sample variance. Despite the number of batches $b \to \infty$ as $n \to \infty$, the computational effort required is still O(n) and O(1) storage.

4.2.3 Increment selection

Glynn and Whitt [58] advised about the lack of consistency of estimators based on a fixed number of batches. It means that sequential CIPs based on batch means should have the number of batches increased as the run length increases. To my knowledge, no one has clearly stated how the number of batches should increase.

Our hypothesis H_3 claims that after selecting the optimal batch size m^*, observations are reorganized into $10 \leq b \leq 30$ batches and stopping rule is checked. Whenever the stopping rule is not achieved, the number b of batches should be

incremented slightly as long as the required precision is not yet achieved. Note that NOBM is a specific case of NOBM/GW when Incr=0.

Let us investigate empirically our hypothesis. In this experiment, we want to construct a confidence interval at 95% of confidence level, when simulating the mean waiting time of clients of queuing systems heavy loaded, e.g. $M/M/1$ and $M/H_2/1$ when traffic intensity is 95%. After grouping the observations according to Schmeiser's findings (b=25), we increased b by **incr** batches (**incr** = 2, 4, and 8), each time the stopping rule failed. The stopping rule is a 5%-relative precision.

We applied the sequential coverage analysis described in section 3.3 : We repeated this sequential simulation over and over, until the relative precision of the coverage was less than or equal 5%. All simulations were executed using strictly nonoverlapping sequences of pseudorandom numbers, generated by a multiplicative congruential generator with multiplier $7^5 = 16807$ and modulus $2^{31} - 1$. Fig. 5 summarizes the experiment for a single processor.

The horizontal dashed line at 0.95 indicates the nominal coverage, that is, the desired confidence level of the final confidence interval. The horizontal dashed line at 0.90 indicates the minimum value of coverage that we can accept (in this research) as reasonable. We call hereafter that region between 0.90 and 0.95 the acceptance region.

For low/medium values of traffic intensity of $M/M/1$, the coverage is close to the confidence level for any value of increment, but as traffic intensity increases, adding more increment seems to yield some coverage erosion. For low/medium values of traffic intensity of $M/H_2/1$, small increment is still better, but for very high traffic intensity there is no difference in choosing small or large increment. Therefore, if we want to be in the the safe side, small increment should be enough to yield asymptotic valid confidence interval.

In terms of run length, there is no significative modification as the increment is increased. The same can be said of the coefficient of variation of the final confidence interval half-width as one can see in Table 3. **CoV{H}** stands for coefficient of variation of the final confidence interval half-width, and **R** stands for how many time the coverage experiment was repeated.

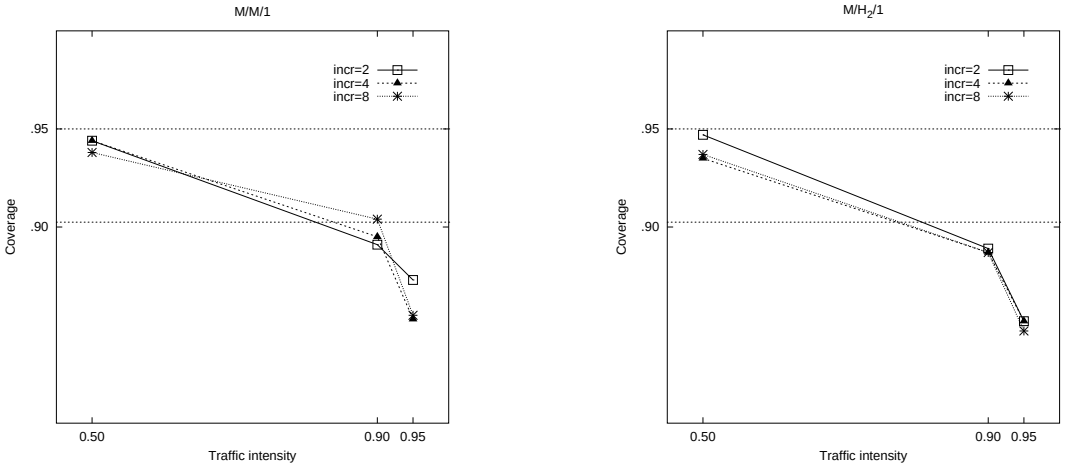


Fig. 5: NOBM/GW : coverage analysis for different values of increment.

		$M/M/1$			$M/H_2/1$		
incr	ρ	run length	CoV$\{H\}$	R	run length	CoV$\{H\}$	R
2	0.50	42.871	0.0412	2554	79.333	0.0273	2708
	0.90	466.885	0.0305	1552	706.628	0.0311	1416
	0.95	1496.79	0.0329	1260	2254.59	0.0325	1077
4	0.50	42.847	0.0420	2690	80.596	0.0280	2655
	0.90	467.206	0.0302	1581	701.725	0.0311	1289
	0.95	1494.36	0.0344	1145	2220.90	0.0345	1016
8	0.50	43.424	0.0432	2614	81.424	0.0288	2700
	0.90	466.827	0.0301	1672	701.785	0.0311	1289
	0.95	1494.95	0.0344	1141	2258.70	0.0332	1048

Tab. 3: NOBM/GW : effect of increment over run length and CoV{H}

Pseudocode 2 Sequential procedure based on NOBM/GW

1: $\mathbf{m} := 100; \mathbf{b} := 100$; **incr** $:= 2$
2: $\boldsymbol{\epsilon} := 0.05; \boldsymbol{\beta} := 0.1$
3: $L := 0.1 * \mathbf{b}$
4: $\beta_k := \beta / L$
5: Uncorrelated = false
6: $n := 0$; $contM := 0$; $contB := 0$; $Bmean := 0$; $t := 1$
Require: Observations are in the stationary phase
7: **while** (**not** Uncorrelated) **do**
8: **repeat**
9: **while** ($contM <$ **t*m**) **do**
10: Collect an observation X
11: $n := n + 1$
12: $contM := contM + 1$
13: $Bmean := Bmean + X$
{This loop remains until collecting a complete batch}
14: **end while**
15: $BM[contB] := Bmean/$**contM**
16: $contB := contB + 1; contM := 0$
{This loop remains until collecting b batches}
17: **until** $(n = b * t * m)$
{Test batch means for correlation at significance level β}
18: Compute correlation coefficients of lags from 1 to L
19: $threshold := z_{1-\beta_k/2}\sqrt{v\hat{a}r(\hat{\rho}_k)}$
20: **for** $k = 1$ to L **do**
21: **if** $\hat{\rho}_k > threshold$ **then**
22: t := t + 1
23: Join each t***m** observations and find their means
24: contB := (t - 1) * contB / t;
25: Save batch means in the first contB positions of BM[ ]
26: $contM := 0$; $Bmean := 0$; Uncorrelated := false; break
27: **else**
28: Uncorrelated := true
29: **end if**
30: **end for**
{This loop remains until batch means can be considered uncorrelated}
31: **end while**
{For estimation phase, $10 \leq b \leq 30$}
32: $b := 25$
33: Form b batches by joining every four batches
34: $t := 4 * t$
35: Compute the sample mean $\bar{X}(n)$ and the half-width confidence interval **H**
36: **while** $(\mathbf{H}/\bar{X}(n) > \epsilon)$ **do**
37: Collect incr*t*m observations
38: $b := b + incr$
39: Form b batches
40: Compute the batch means
41: Compute $\bar{X}(n)$ and **H**
42: **end while**
43: StopSimulation = yes

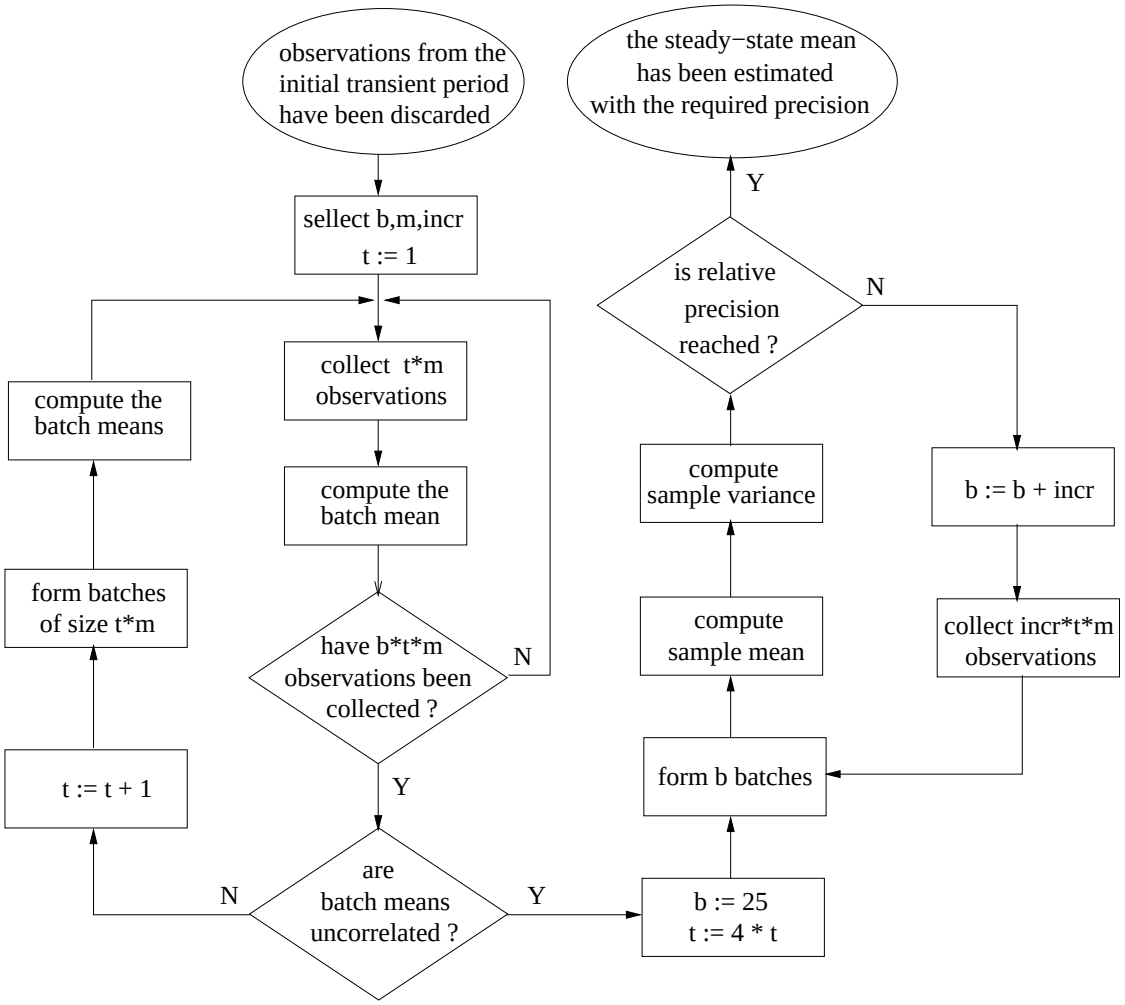


Fig. 6: Flowchart of a sequential procedure based on NOBM/GW.

4.3 Design of a sequential procedure based on SBM

Pseudocode 3 presents a proposal of a sequential run length control procedure based on SBM, and Fig. 7 is the corresponding flowchart. The variables are explained in Tab. 4 :

m	initial batch size indicated by the user;
b	number of batches indicated by the user; due the jackknife estimators, this number must be at least 100;
$\boldsymbol{\epsilon}$	the relative precision requested by the user;
L	number of autocorrelation coefficients;
$\boldsymbol{\beta}$	overall significance level of L tests against correlation;
$\boldsymbol{\beta}_k$	significance level of each of the individual tests against correlation;
contM	counts the observations inside each batch;
contB	counts the number of batches;
n	sample size;
t	grouping factor; batch size will increase as a multiple of the initial batch size;
Uncorrelated	boolean variable indicating whether the batch means can be considered uncorrelated;
s	amount of observations to be discarded. We adopted 20% of the initial batch size.
Incr	number of additional batches to be added at consecutive checkpoints whenever the stopping rule is not achieved;
threshold	reference value for accepting that correlation among batch means is negligible;
$\boldsymbol{z_{1-\beta_k/2}}$	upper $(1-\beta_k/2)$ critical point of the standard normal distribution.
$\hat{\rho}_k$	correlation coefficient of lag k;
Bmean	accumulates observations inside a batch;
BM	array containing the batch means;

Tab. 4: Variables of the pseudocode for SBM.

Each replication executing SBM, like the previous CIPs, uses an single-pass algorithm. Though we have decided for increasing the number of batches while the simulation run length increases, and discard some observations between consecutive groups of m observations, the computational effort required is still O(n) and O(1) storage.

Whenever the test for independence of the batch means fails, the procedure groups contiguous batches but does not fill the gaps resulted from discarding. This is a design decision based on the fact that either one should save that discarded observations and then restore them after unsuccessful tests, or one should fill the gaps with new observations before collecting more observations. In our point of view, both approaches should increase unnecessarily the complexity of the algorithm, and collecting more observations instead should be computationally more efficient. Additionally, by filling gaps with new observations (collected in future time intervals,

thus at different time than earlier removed observations) would lead to random mixing of the order of analyzed time series. Potentially, it can introduce causality errors in the analyzed samples of output data. The statistical properties arising from this decision will be here investigated.

Therefore, initiating with a batch size m, the last s observations of each batch are discarded, yielding a batch of size $(m - s)$. If the test for independence among the batch means fails, observations are grouped in such a way that the next batch size is $2(m - s)$. Observations continue being collected as a multiple of m, always discarding the last s ones. In case of a new unsuccessful test, the next batch size would be $3(m - s)$, and so on.

This is a trial to weaken even more the correlation structure, and to reduce the complexity of a sequential procedure. Otherwise, one should :

- save the discarded observations;
- joint them back to each batch whenever the test for independence fails;
- calculate batch means before continuing collecting more observations;
- joint each t batches;
- collect more observations;
- and so on.

What we expect to have with our proposed procedure are batches with smaller size compared to NOBM, but also less correlated than the original series. It differs from the original proposal in the sense that the authors suggest discarding observations after collecting a batch of size m.

An analogy with our proposal shows that, if the first test for independence among the batch means fails, the original SBM groups batches in such a way that the next batch size is $2m - s$. Observations continue being collected as a multiple of m, always discarding the next s observations. In case of a new unsuccessful test the next batch size would be $3m - 2s$, and so on.

4.3.1 SBM with increasing number of batches

When applying SBM, should we increase the number of batches after each checkpoint fails in the test of relative precision? We simulated an $M/M/1$ to construct a confidence interval at 95% of confidence level, when estimating the mean waiting time of its clients.

At first, we considered no increment of batches at all, and repeated each experiment R times for different values of ρ. The number R of repetitions is obtained sequentially accordingly to the sequential analysis described in section 3.3. In the sequence, we assumed that Glynn and Whitt's findings should be followed, and we increased the number of batches as we did in NOBM/GW. We adopted a spacing that had given the most promising results (in terms of coverage) during pilot runs.

SBM	ρ	$cov \pm H_{cov}$	run length	CoV$\{H\}$	R
no incr	0.10	0.931 ± 0.010	62.085	0.0576	2479
	0.50	0.933 ± 0.010	55.864	0.0540	2484
	0.90	0.906 ± 0.015	462.073	0.0399	1585
	0.95	0.853 ± 0.022	1494.36	0.0344	1153
with incr	0.10	0.935 ± 0.009	53.121	0.0303	2697
	0.50	0.939 ± 0.009	48.703	0.0343	2738
	0.90	0.913 ± 0.014	472.768	0.0297	1683
	0.95	0.876 ± 0.019	1505.77	0.0335	1243

Tab. 5: SBM with and without increasing number of batches

Table 5 makes it clear that SBM with increment yields coverage a little bit better, always requires less observations to yield a final result, and the confidence intervals are more stable. That's our choice.

A rule of thumb derived from the observation of the experiments of this research showed us that when a CIP (or certain configuration of a CIP) performs better than another one (or better than other configuration of the same CIP), the sequential coverage analysis requires more repetitions to yield a coverage with the required precision. Note that R for SBM with no increment is consistently lower than the corresponding for SBM with increment. That means, SBM with increment of batches is better than SBM with no increment. The direct comparison of the main MOE's (coverage, run length and CoV{H}) confirm that conclusion.

4.3.2 Spacing selection

If one could guess in advance how should be the magnitude of the spacing **s** suitable to the underlying process being simulated, it would save computation time to stablish this quantity and, equivalently, shorten the batch size selection phase.

We investigated the effect on coverage when we select **s** as a percentage of the initial batch size m given by the user. Table 6 illustrates the performance of this approach, when the mean waiting time of a 95%-loaded M/M/1 and $M/H_2/1$ queues were simulated by using **s**=0.1m, 0.2m, 0.3m and 0.4m.

By simulating $M/M/1$, coverage was roughly equal for all selected values of spacing. Spacings 0.2m and 0.4m required less observations than the other two, but the former produced more stable confidence intervals.

By simulating $M/H_2/1$, some differences in coverage did appear, and spacing 0.2m seems to be more promising. Additionally, this spacing required less observations, but stability of confidence intervals were practically identical. Therefore, 0.2m will be used as the spacing in our empirical investigation of SBM performance.

Pseudocode 3 Sequential procedure based on SBM

```
m := 100; b := 100; incr := 2
s := 0.2 * m
ϵ := 0.05; β := 0.1
L := 0.1 * b
β_k := β/L
Uncorrelated := false
n := 0; contM := 0; contB := 0; Bmean := 0; t := 1
Require: Observations are in the stationary phase
while (not Uncorrelated) do
  repeat
    while (contM < t*m) do
      Collect an observation X
      n := n + 1
      contM := contM + 1
      if contM ≤ (m − s) then
        Bmean := Bmean + X
      else
        Discard this observation
      end if
      {This loop remains until collecting a complete batch}
    end while
    BM[contB] := Bmean/(m − s)
    contB := contB + 1; contM := 0
    {This loop remains until collecting b batches}
  until (n = b * t * m)
  {Test batch means for correlation at significance level β}
  Compute correlation coefficients of lags from 1 to L
  threshold := z_{1−β_k/2} sqrt(v̂ar(ρ̂_k))
  for k = 1 to L do
    if ρ̂_k > threshold then
      t := t + 1; contB := (t - 1) * contB / t;
      Join each (t*m - s) observations and find their means
      contM := 0; Bmean := 0
      Uncorrelated := false; break
    else
      Uncorrelated := true
    end if
  end for
  {This loop remains until batch means can be considered uncorrelated}
end while
{For estimation phase, 10 ≤ b ≤ 30}
b := 25; t := 4 * t
Form b batches by joining every four batches
Compute X̄(n) and H
```

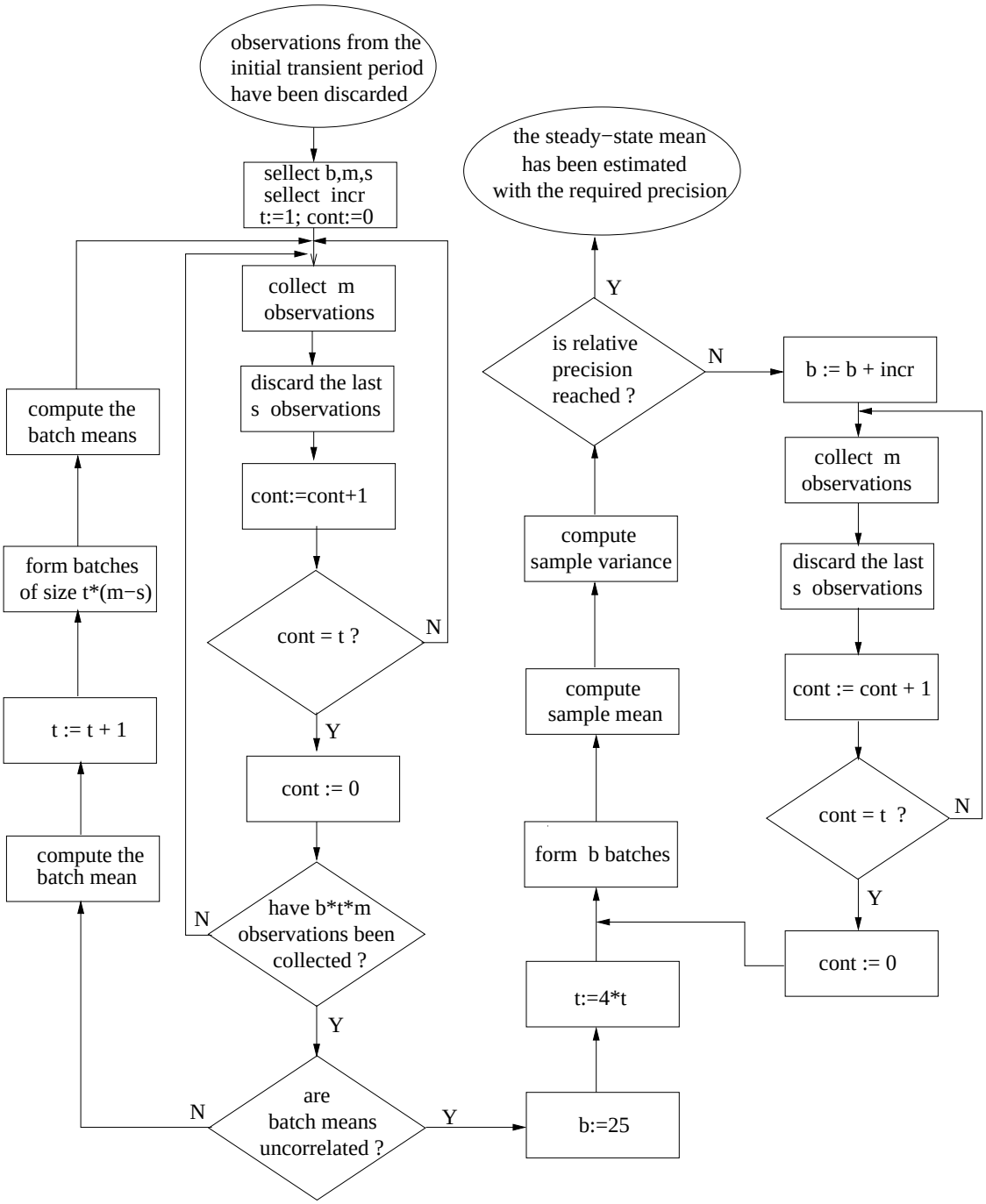


Fig. 7: Flowchart of a sequential procedure based on SBM.

Sequential procedure based on SBM
while $(\mathbf{H}/\bar{X}(n) > \epsilon)$ **do**
  $b := b + incr$
  repeat
    Collect an observation X
    $n := n + 1; contM := contM + 1$
    if $contM \leq (\mathbf{m} - \mathbf{s})$ **then**
      $Bmean := Bmean + X$
    else
      Discard this observation
    end if
    $BM[contB] := Bmean/(\mathbf{m} - s)$
    $contB := contB + 1; contM := 0$
  until $(n = b * t * m)$
  Form b batches
  Compute $\bar{X}(n)$ and $\mathbf{H}$
end while
StopSimulation = yes

queue	spacing	ρ	η	$cov \pm H_{cov}$	run length	CoV{H}	R
$M/M/1$	0.1m	0.95	0.95	0.880 ± 0.019	1519.01	0.0324	1256
	0.2m	0.95	0.95	0.876 ± 0.019	1505.77	0.0335	1243
	0.3m	0.95	0.95	0.869 ± 0.020	1520.51	0.0331	1236
	0.4m	0.95	0.95	0.876 ± 0.019	1498.34	0.0378	1238
$M/H_2/1$	0.1m	0.95	0.95	0.848 ± 0.022	2244.35	0.0337	1089
	0.2m	0.95	0.95	0.862 ± 0.021	2206.10	0.0330	1143
	0.3m	0.95	0.95	0.850 ± 0.023	2262.55	0.0328	1078
	0.4m	0.95	0.95	0.852 ± 0.021	2262.28	0.0335	1152

Tab. 6: Spacing selection for SBM

4.4 Design of a sequential procedure based on OBM

Pseudocode 4 presents a proposal of a sequential run length control procedure based on OBM, and Fig. 8 is the corresponding flowchart. The variables are explained in Tab. 7 :

Although overlapped batch means are indeed dependent, we use the same heuristic to find the optimal batch size $m*$ as in the classical estimator, and apply Schmeiser's suggestion and apply the overlapping algorithm, with each sub-batch initiating a new overlapped batch.

After collecting $n = b.m$ observations, the procedure computes so many batch means as is the number of overlapped batches. To save storage, we maintain in memory just the last batch. As a new observation arrives it is appended at the end of the batch while the first observation of the batch is deleted. This mechanism is generically known as *+1-1*. A counter advises when the degree of overlapping is achieved, which means that a batch mean has to be calculated. The procedure goes on until the stopping rule is met.

m	initial batch size indicated by the user;
b	number of batches indicated by the user; due the jackknife estimators, this number must be at least 100;
$\boldsymbol{\epsilon}$	the relative precision requested by the user;
L	number of autocorrelation coefficients;
$\boldsymbol{\beta}$	overall significance level of L tests against correlation;
$\boldsymbol{\beta}_k$	significance level of each of the individual tests against correlation;
contM	counts the observations inside each batch;
contB	counts the number of batches;
n	sample size;
t	grouping factor; batch size will increase as a multiple of the initial batch size;
Uncorrelated	boolean variable indicating whether the batch means can be considered uncorrelated;
threshold	reference value for accepting that correlation among batch means is negligible;
$\boldsymbol{z_{1-\beta_k/2}}$	upper $(1-\beta_k/2)$ critical point of the standard normal distribution.
$\hat{\rho}_k$	correlation coefficient of lag k;
Bmean	accumulates observations inside a batch;
BM	array containing the batch means;
doo	degree of overlapping;

Tab. 7: Variables of the pseudocode for OBM.

Higher degrees of overlapping implies in larger number of batch means to be computed, which also means more values are used in the estimation. Statistically, it implies in lower variance and, expected better coverage.

4.4.1 Degree of overlapping

If one divides each batch into two equal sized parts, each one with $m^*/2$ observations, in such a way that each part initiates a new (overlapped) batch of size m^*, one can form $2(b-1)+1$ batches. By dividing each batch in four equal sized parts one obtains $4(b-1)+1$ batches, and so on (see Table 8).

The last case (m^*/m^*) is known as *complete overlap*, that is each observation begins a new (overlapped) batch. Meketon and Schmeiser [89] suggested an asymptotic number of freedom equal $1.5(b-1)$. The other cases are generally known as *partial overlap*, and the associated degree of freedom for different degrees of overlapping can be found by applying the results of Welch [131], when he investigated the relationship between overlapping batch means and the spectral estimator.

However, to get a 1/3 variance reduction when compared to the variance of the classical NOBM, Meketon and Schmeiser let $b \to \infty$ [113], which means that prob-

Pseudocode 4 Sequential procedure based on OBM

1: $\mathbf{m} := 100; \mathbf{b} := 100;\ \boldsymbol{\epsilon} := 0.05; \boldsymbol{\beta} := 0.1$
2: $L := 0.1 < *\mathbf{b}$
3: $\beta_k := \beta / L$
4: Uncorrelated := false
5: $n := 0$; $contM := 0$; $contB := 0$; $Bmean := 0$; $t := 1$
Require: Observations are in the stationary phase
6: **while** (**not** Uncorrelated) **do**
7: **repeat**
8: **while** ($contM < t * \mathbf{m}$) **do**
9: Collect an observation X
10: $n := n + 1$
11: $contM := contM + 1$
12: $Bmean := Bmean + X$
{This loop remains until collecting a complete batch}
13: **end while**
14: $BM[contB] := Bmean/\mathbf{m}$; $contB := contB + 1$; $contM := 0$
{This loop remains until collecting b batches}
15: **until** ($n = b * t * m$)
{**Test batch means for correlation at significance level** β}
16: Compute correlation coefficients for lags from 1 to L
{$threshold := z_{1-\beta_k/2}\hat{\sigma}[\hat{\rho}_k]$}
17: **for** $k = 1$ to L **do**
18: **if** $\hat{\rho}_k > threshold$ **then**
19: t := t + 1; contB := (t - 1) * contB / t;
20: Join each t*$\mathbf{m}$ observations and find their means
21: Save batch means in the first contB positions of BM[ ]
22: $contM := 0$; $Bmean := 0$; Uncorrelated := false; break
23: **else**
24: Uncorrelated := true
25: **end if**
26: **end for**
{This loop remains until batch means can be considered uncorrelated}
27: **end while**
{For estimation phase, $10 \leq b \leq 30$}
28: $b := 25$
29: Form b batches
30: $t := 4 * t$
31: Compute batch means for all (n - t*m + 1) possible batches
32: Save the observations of the last batch
33: Compute $\bar{X}(n)$ and $\mathbf{H}$
34: Compute $\mathbf{H}$
35: **while** ($\mathbf{H}/\bar{X}(n) > \epsilon$) **do**
36: Collect an observation X
37: Add X to the end of the saved batch
38: Delete the first observation of the saved batch
39: Compute $\bar{X}(n)$ and $\mathbf{H}$
40: **end while**
41: StopSimulation = yes

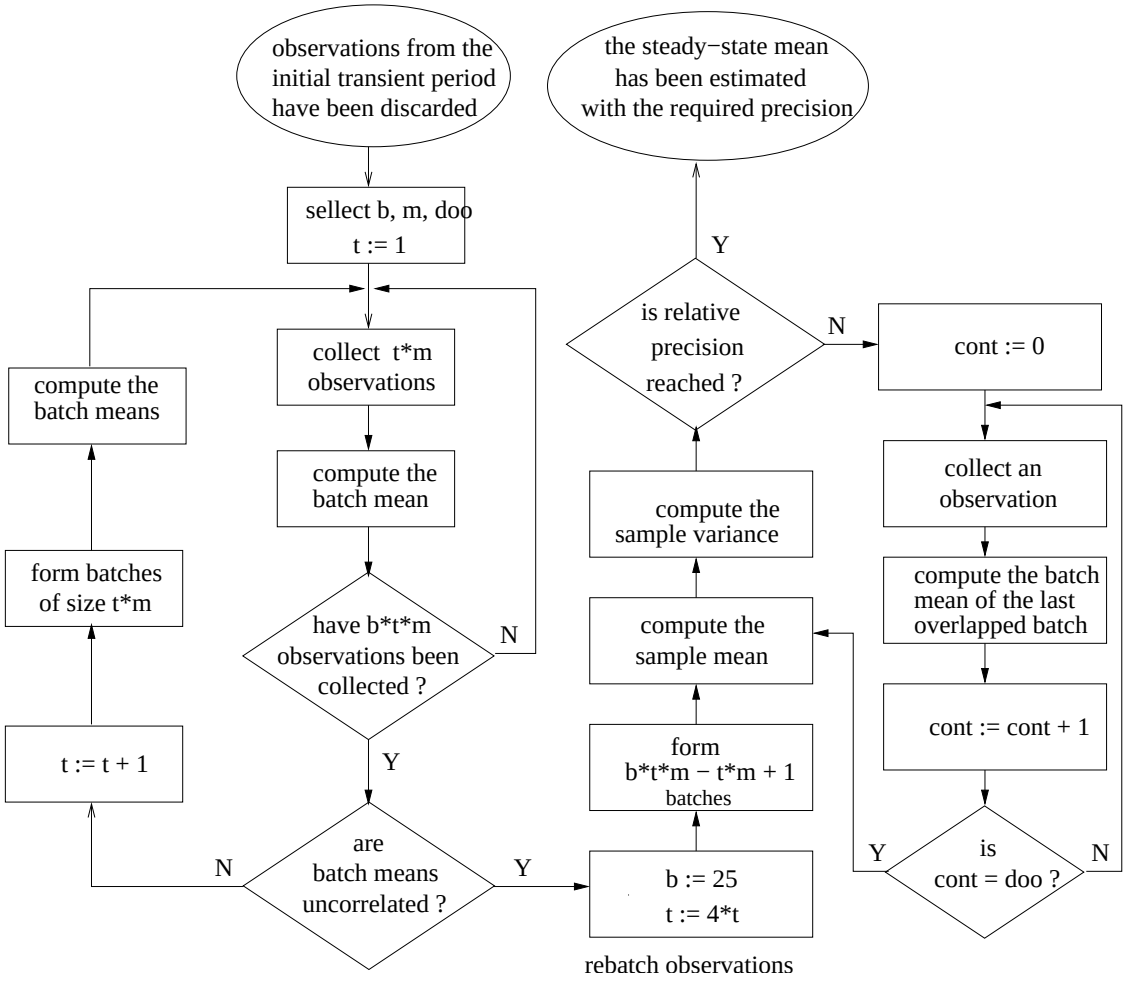


Fig. 8: Flowchart of a sequential procedure based on OBM.

ably those differences in degrees of freedom are practically irrelevant in terms of sequential procedure, especially for very high levels of traffic intensity.

Results for simulation of M/M/1 with P=1, shown in Table 4.5, support this statement. For just one reason we have chosen to use the complete overlapping : it offers greater flexibility for achieving better speedup under MRIP, since we can control the granularity of checkpoints.

Each replication executing OBM uses a single-pass algorithm, but to achieve computational complexity O(n), OBM requires saving the previous m observations, and each new batch mean is obtained by adding and subtracting one observation

Degree of overlap	Number of batches	degrees of freedom
$m^*/2$	$2(b-1)+1$	1.33
$m^*/4$	$4(b-1)+1$	1.45
$m^*/8$	$4(b-1)+1$	1.48
...	...	...
m^*/m^*	$m(b-1)+1$	1.50

Tab. 8: Degrees of freedom for OBM, following Welch [131]

from the previous batch size, so it needs O(m) storage [61].

Degree of overlap	ρ	η	$cov \pm H_{cov}$	run length	CoV{H}
$m^*/2$	0.95	0.95	0.949 ± 0.011	2145.911	0.0257
$m^*/4$	0.95	0.95	0.928 ± 0.014	2129.494	0.0270
$m^*/8$	0.95	0.95	0.940 ± 0.012	2131.604	0.0268
m^*/m^*	0.95	0.95	0.949 ± 0.010	2212.278	0.0255

Tab. 9: OBM performance for different degrees of overlapping

4.5 Design of a sequential procedure based on STS

Pseudocode 5 presents a proposal of a sequential run length control procedure based on STS/CSUM, and Fig. 9 is the corresponding flowchart. The variables are explained in Tab. 10 :

m	initial batch size indicated by the user;
b	number of batches indicated by the user; due the jackknife estimators, this number must be at least 100;
$\boldsymbol{\epsilon}$	the relative precision requested by the user;
L	number of autocorrelation coefficients;
$\boldsymbol{\beta}$	overall significance level of L tests against correlation;
$\boldsymbol{\beta}_k$	significance level of each of the individual tests against correlation;
contM	counts the observations inside each batch;
contB	counts the number of batches;
n	sample size;
t	grouping factor; batch size will increase as a multiple of the initial batch size;
A_i	random variables associated with ith batch;
isNormal	indicates whether (or not) the random variables A_i's can be considered as being normally distributed;
S^2	temporary variables to be used in the Shapiro-Wilk test;
v	temporary variables to be used in the Shapiro-Wilk test;
W	the Shapiro-Wilk statistic;
$\mathbf{W}_\alpha$	threshold of the Shapiro-Wilk test;

Tab. 10: Variables of the pseudocode for OBM.

To estimate a confidence interval **I** for the steady-sate parameter $\mu = E[X_i]$, such that $\mu \in \mathbf{I}$ with probability η, one should :

- guarantee, as accurate as possible, the elimination of initialization bias
- group the observations into b adjacent, nonoverlapping equal-sized batches with m observations in each batch ($n = b.m$).
- center the output series to have a mean equal zero :

$$S_i(k) = \bar{X}_{i,m} - \bar{X}_{i,k}, \tag{4.4}$$

where $\bar{X}_{i,k}$ is the average of the first k observations of batch i ($k = 1, \ldots, m, S_i(0) = 0$).

- scale the magnitude of the sequence, dividing $S_i(k)$ by $\sqrt{M}\sigma/k$. Here the constant σ is given by :

$$\sigma^2 = \sigma_y^2 + 2\sum_{i=1}^{\infty} \gamma_y(i) = lim_{n\to\infty} nVar(\bar{X}(n)) \tag{4.5}$$

with

$$\sigma_y^2 = Var(X_i)$$

and

$$\gamma_y(i) = Cov(X_0, X_i)$$

- scale the series index to the unit interval t=k/M, yielding the standardized time series :

$$T_i(t) = \frac{[mt]S_i([Mt])}{\sigma\sqrt{m}}, \quad 0 \leq t \leq 1, \tag{4.6}$$

where [.] denotes the greatest integer function.

- apply a central limit theorem to obtain a probability model for the behavior of the sequence $T_i(t)$. This probability model converges asymptotically to a standard Brownian bridge process.
- use the properties of the limiting Brownian bridge process to construct a confidence interval for the original time series.

Pseudocode 5 Sequential procedure based on STS

1: $\mathbf{m} := 100$
2: $\mathbf{b} := 10$
3: isNormal = no
Require: Observations are in the stationary phase
4: **while** (**not** isNormal) **do**
5: Collect $\mathbf{n} = \mathbf{b} * \mathbf{m}$ observations
{Compute A_i for each batch}
6: **for** $i = 1$ to $\mathbf{b}$ **do**
7: **for** $j = 1$ to $\mathbf{m}$ **do**
8: $A_i = \sum_{j=1}^{m}[(\mathbf{m}+1)/2 - j]X_{(i-1)m+j}$
9: **end for**
10: **end for**
{Test A_i's for normality by applying Shapiro-Wilk test}
{$H_o : A_i$'s are normally distributed}
11: Compute $S^2 = \sum_{i=1}^{b} {A_i}^2 - 1/b(\sum_{i=1}^{b} A_i)^2$
12: Calculate $v = \sum_{i=1}^{k} a_{b-i+1}(X_{b-i+1} - X_i), k = b/2$
{a_{b-i+1} are the tabled S-W coefficients; see Appendix A}
{S-W statistic}
13: $W = v^2/S^2$
{α is the percentage point of the test; see Appendix A}
14: **if** $W < W_\alpha$ **then**
15: the null hypothesis H_o is rejected
16: **end if**
17: **end while**
18: Compute the point estimator : $\bar{X}_n = 1/n \sum_{i=1}^{n} X_i$
19: $\hat{V}_T = \frac{12}{(m^3-m)} \sum_{i=1}^{b} {A_i}^2$
20: Compute $\mathbf{H} = t_{2b-1,1-\frac{\alpha}{2}} \sqrt{\frac{\hat{V}_T}{n}}$
{ϵ is the desired relative precision}
21: **while** $(\frac{H}{\bar{X}_n} > \epsilon)$ **do**
22: Increase $\mathbf{m}$
23: Collect $\mathbf{n} = \mathbf{b} * \mathbf{m}$ observations
24: Compute A_i for each batch
25: Compute the point estimator : $\bar{X}_n = 1/n \sum_{i=1}^{n} X_i$
26: Compute $\mathbf{H} = t_{2b-1,1-\frac{\alpha}{2}} \sqrt{\frac{\hat{V}_T}{n}}$
27: **end while**
28: StopSimulation = yes

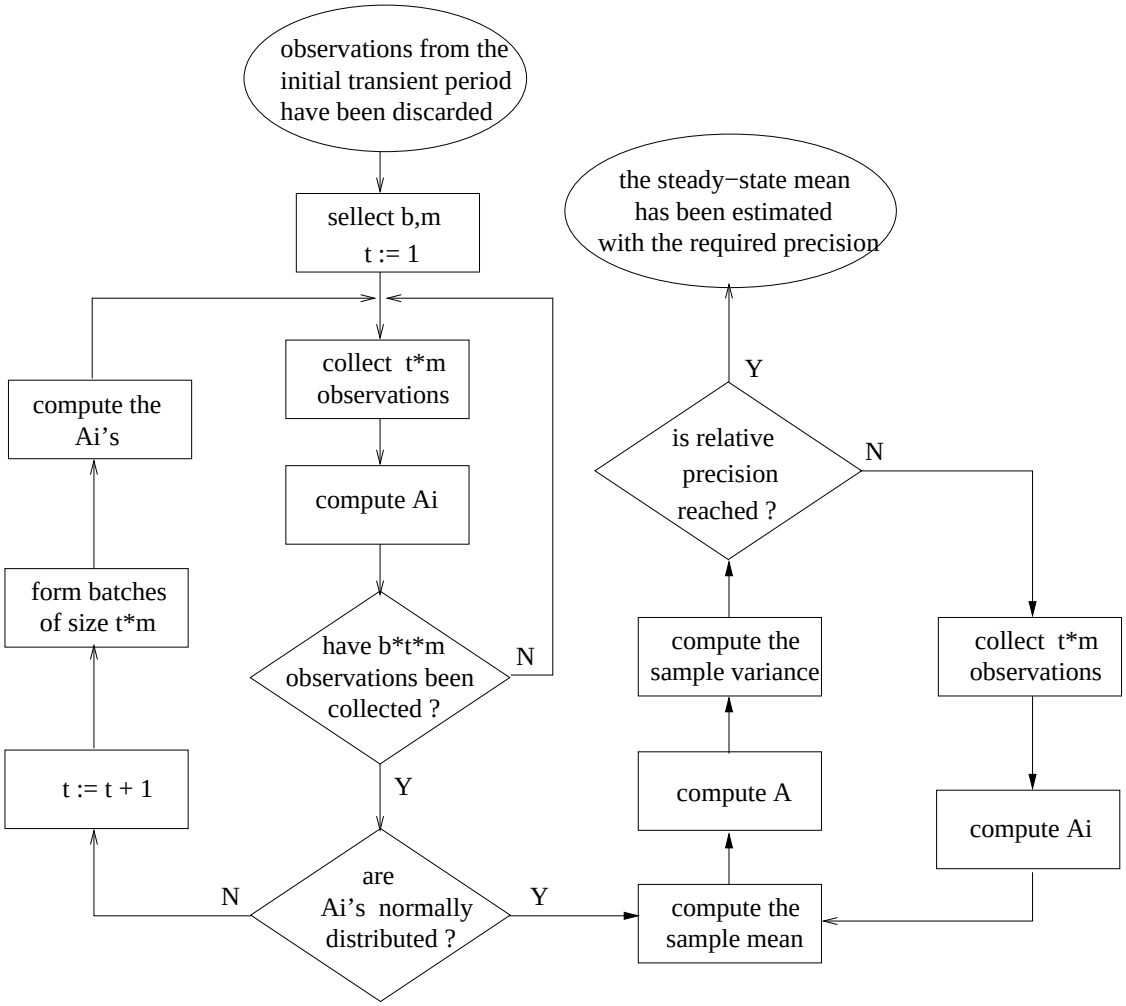


Fig. 9: Flowchart of a sequential procedure based on STS/CSUM.

Schruben [119] and Goldsman/Schruben [62] developed some estimators based on different characteristics of the Brownian bridge process. After grouping the observations into b batches of size m, and computing the mean of the ith batch as being $\bar{X}_i(m) = m^{-1}\sum_{i=1}^{m} X_i$ and the grand mean $\bar{X}(n) = b^{-1}\sum_{i=1}^{b} \bar{X}_i(m)$, the classical interval estimator at a confidence level $(1-\alpha)$ is given by $I_{class} = [\bar{X}(n) \pm H_{class}]$, with

$$H_{class} = t_{b-1,1-\frac{\alpha}{2}} \sqrt{\sum_{i=1}^{b} \frac{(\bar{X}_i(m) - \bar{X}(n))^2}{b(b-1)}}$$

where $t_{b-1,1-\frac{\alpha}{2}}$ is the upper $(1-\frac{\alpha}{2})$ critical point for the t distribution with b-1 degrees of freedom at $1-\alpha$ confidence level.

After standardizing each observation one can find the random variables A_i, the asymptotic scaled sum of $T_i(t)$, for each batch by means of

$$A_i = \sigma\sqrt{m}\sum_{k=1}^{m} T_i(t) = \sum_{k=1}^{m}\sum_{j=1}^{k}(\bar{X}_{j,m} - \bar{X}_{j,k}) \tag{4.7}$$

A simplification can be found toward facilitating a sequential procedure. From (2.21) and (2.20) one can find :

$$\begin{aligned} A_i &= \sigma\sqrt{m}\sum_{k=1}^{m}\Big[\frac{k(\bar{X}_{i,k} - \bar{X}_{i,m})}{\sigma\sqrt{m}}\Big] \\ &= \sum_{k=1}^{m} k\Big[\frac{1}{k}\sum_{j=1}^{k} X_{(i-1)m+j} - \frac{1}{m}\sum_{j=1}^{m} X_{(i-1)m+j}\Big] \\ &= \sum_{k=1}^{m}\Big[\sum_{j=1}^{k} X_{(i-1)m+j} - \frac{k}{m}\sum_{j=1}^{m} X_{(i-1)m+j}\Big] \end{aligned}$$

$$A_i = \sum_{k=1}^{m}\Big[j - \frac{m+1}{2}\Big]X_{(i-1)m+j} \tag{4.8}$$

Then, for sufficiently large m the random variables A_i become approximately IID normal. By computing the statistic

$$A = \sum_{i=1}^{b}\frac{12{A_i}^2}{(m^3 - m)} + m(\bar{X}_i - \bar{X})^2$$

an asymptotically valid combined **c**lassical-sum interval estimator $[\bar{X} \pm \mathbf{H}]$ can be constructed for performance parameter μ, considering that

$$V_{csum} = \frac{A}{2b-1} \tag{4.9}$$

and

$$\mathbf{H} = t_{2b-1,1-\alpha/2}\sqrt{\frac{A}{2b-1}} \tag{4.10}$$

The computational effort required in standardizing a time series is roughly equal to summing the original time series twice [119], that is, STS/CSUM requires O(n) computation and O(1) storage.

4.5.1 Variants

The problem of testing normality is basic to much statistical theory and practice. Many tests to check the adequacy of the assumption of normality have been proposed. Good references are [27] and [109].

Studies such as the Monte Carlo study of Shapiro and others [123], using a range of populations and sample sizes, have consistently shown that for testing goodness-of-fit of normal distributions, the Shapiro-Wilk statistic has superior power to other statistics for testing a complete sample for normality. Fishman [40] pointed out some other interesting properties of this test.

As we intend to apply such test in sequential procedures, we should take into account that it can be repeated many times until the incoming data can be considered normally distributed. Thus, each part of this test has to be as simple and efficient as possible.

Concerning procedures based on STS, asymptotic arguments found in [120], show that the batch size should grow as the sample size increases. The common method is to fix the number of batches, but keeping it not so large. Schruben [120] suggests using 10 or 20 batches, similar to the results of Schmeiser. Keeping in mind that choosing a small number of batches means leads earlier to the test for normality, we decided to work with 10 batches.

Remember from section 3.5.4 that the Shapiro-Wilk test for normality requires that the sequence be sorted in increasing order. By choosing a small number of batches we do not need a sophisticated sorting procedure, but a simple one such as bubble sort, where each phase of the algorithm consists of "passing through" the unsorted portion of the sequence, by comparing adjacent values and changing their positions that are out of order.

In fact, we implemented two versions of CSUM. Taking into account that after selecting m^*, every time the relative precision fails the procedure must collect more observations and recalculate the A_i's, and this can be time-consuming. The first version (that will used as reference for the empirical investigation in the next chapter), hereafter called CSUM.1, collects a entire new batch of observations and increase b. It has been demonstrated to accelerate the procedure.

The second version, hereafter called CSUM.2, was implemented towards the speedup issues we discuss in the following chapter. It follows Schruben's suggestion and fix the number of batches. The difference is that the procedure does not need to collect a new batch every time the stopping condition test fails. Instead, the procedure can collect a number of observations that is multiple of b. This simple artefact improves considerably the granularity of checkpoints.

4.6 Summary

This chapter presented the design of sequential batch-means-based CIPs we investigated in this research. All of them apply a stationarity test for detecting the end of the transient period, and then they discard the observations collected until then.

NOBM/GW, SBM and OBM select the (optimal) batch size m^* by increasing the batch size of 100 batches, and test batch means against correlation. Every m observations collected, SBM discards systematically the last s observations. At least for the queuing systems here investigated, we found reasonable to select s equal to 20% of the initial batch size given by the user. When m^* is selected, observations are grouped into 25 batches. STS/CSUM finds m^* by testing against normality the random variables A_i's of 10 batches.

In the estimation phase, NOBM/GW and SBM collect two new batches of size m^* whenever the stopping rule is checked and fails. OBM checks the stopping rule whenever a number of observations equal to the degree of overlapping is collected. STS/CSUM checks the stopping rule whenever a multiple of 10 observations is collected.

5

Experimental investigation

5.1 Introduction

Methodology of a good experimental design helps make the empirical results more applicable to real situations. Each proposed CIP is based on assumptions about the random characteristics of output data. Provided that the output stochastic process is covariance stationary :

- Batch Means based procedures rely on the fact that the means of the batches become approximately normally distributed and uncorrelated, and therefore independent, as the batch size increases;
- Standardized time series based procedures additionally assumes that the output stochastic process is $\phi - mixing$, i.e., the correlations of the stochastic process output sequence die off, and that for a sufficiently large batch size the random variables A_i (see Equation2.21) associated with each batch, become approximately IID normal.

The correctness of these assumptions depends on how the data is generated, collected, and processed [117].

We used a set of reference queuing models explained in Appendix C for determining, experimentally, the behavior of the proposed sequential batching procedures to be run under MRIP. They represent simplified versions of models (with analytically tractable solutions) that often arise in simulations of computer communication systems. We fixed the mean service time μ to be 10.0 and varied the mean interarrival time λ from 1.0 to 9.5.

The starting conditions of each simulation experiment are empty queue and idle service. Generally, by simulating real-world problems it would be difficult to determine the starting conditions from pilot runs, such as suggested in [71], though it

could reduce the problems associated with the transient period. We attacked the initial transient problem by applying the Schruben's test [121].

We ran an exhaustive set of simulation experiments to estimate the expected average waiting time for the customers of the selected queuing systems and, since $\rho < 1$, there exists a steady-state random variable W_n such that $W_n \rightarrow W$ as $n \rightarrow \infty$. The estimators were obtained with the relative precision of 0.05, and all statistical tests were performed at the significance level of 0.05. Unless stated otherwise, results presented in this chapter were obtained by means of the sequential coverage analysis explained in section 3.3.

We are going to present important statistical properties of these sequential batching procedures, at first using one processor and then to investigate improvements offered by MRIP. Experimental analysis of coverage is required for assessing the quality of practical implementations of methods used for determining confidence intervals in steady-state simulation. The reference set of queuing models used in our empirical investigation is explained in Appendix C.

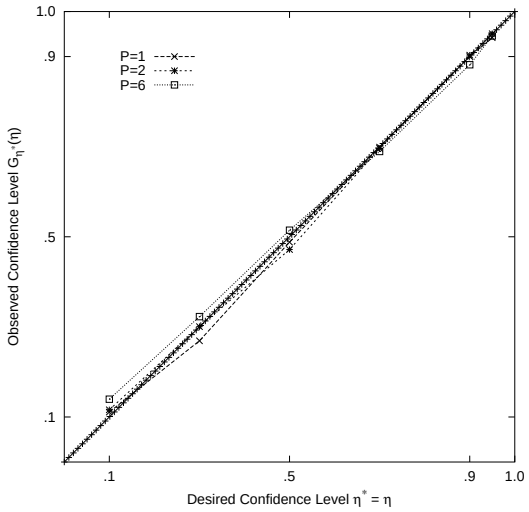
5.2 General overview

Concerning the behavior of the analyzed CIPs, when both the coefficient of variation of service mechanism C_x traffic intensity are low, we observe that adding more processors is of little help, since coverage is already acceptable and it is not possible to reduce the run length to improve this result. Figure 10 shows the coverage function for NOBM/GW, SBM, OBM and STS/CSUM, when we simulated the M/D/1 queue with 1, 2, and 6 processors, and traffic intensity 0.10.

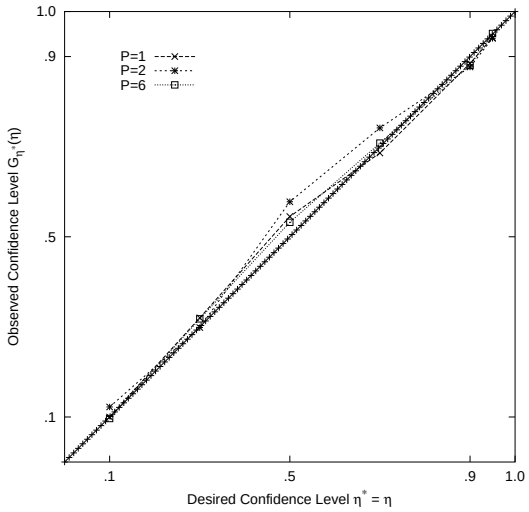
By increasing traffic intensity to $\rho = 0.50$, the classical NOBM produces an oscillating coverage function, depicted in Figure 11, as more processors take part in the simulation of M/D/1 : for small confidence levels the coverage function is high; for large confidence levels the coverage function is low. This behavior can be undesirable, since under this scenario the procedure is not efficient at all (either wasteful or insufficient). Under MRIP more sophisticated procedures, such as SBM, OBM, and STS/CSUM, eliminate this behavior.

As long the traffic intensity increases, correlation among observations becomes stronger. MRIP is of much help, as it contributes to improve the coverage and reduce the average run length. Figure 12 emphasizes that this claim is true only for robust methods. While by NOBM the coverage when simulating $M/E_4/1$ is closer to the confidence level, adding more processors does not change the average run length necessary to yield results with the desired accuracy. By STS/CSUM, on the contrary, when simulating $M/E_4/1$, coverage improves as the number of processors increases, and one can perceive a trend to collect fewer observations processors. In Appendix H the reader can verify that this trend also occurs for $M/D/1$ and $M/M/1$.

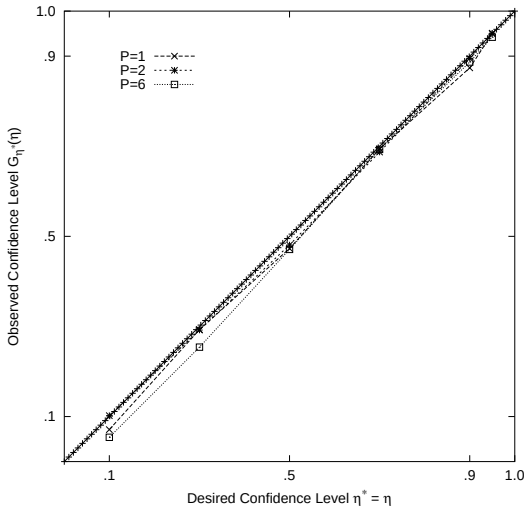
The benefits of MRIP can be better evaluated when traffic intensity is high, and the method of analysis is robust, as we will show in next sections. Increasing



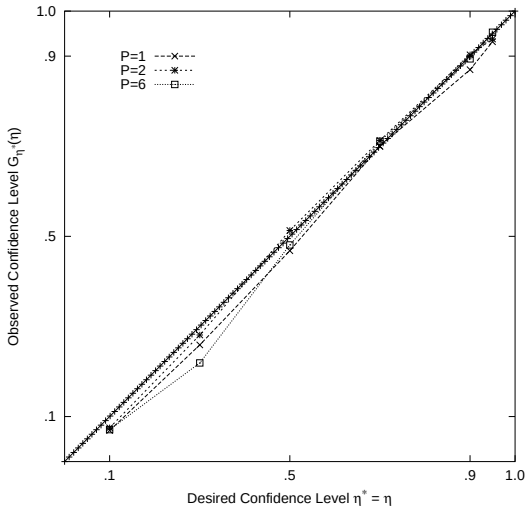
(a) NOBM/GW



(b) SBM

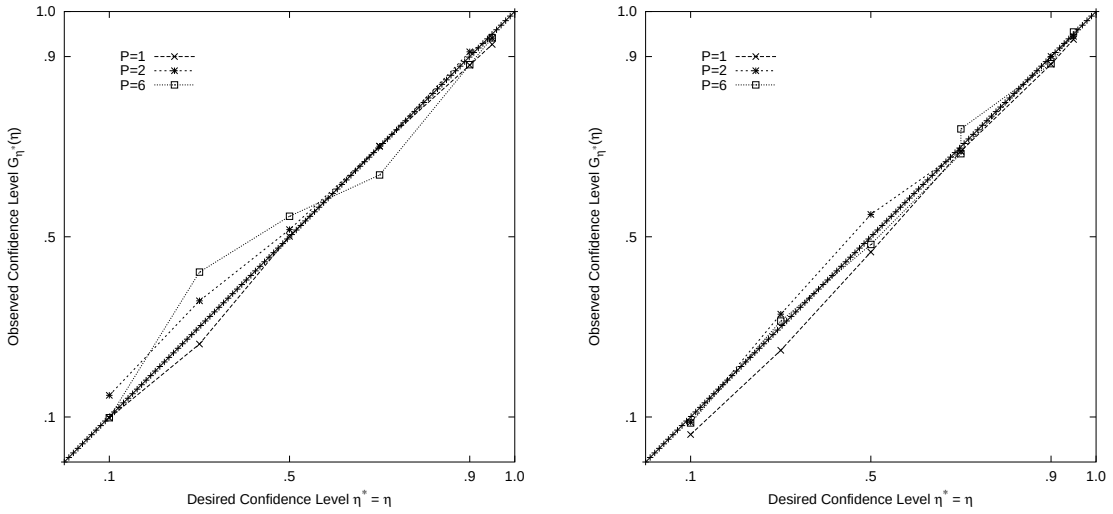


(c) OBM



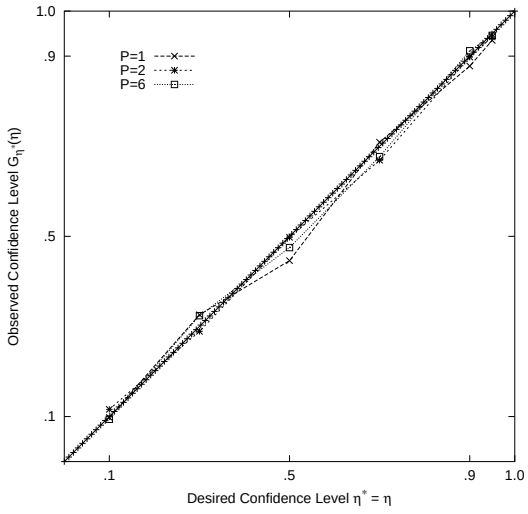
(d) sTS/CSUM

Fig. 10: Coverage function for low coefficient of variation and $\rho = 0.10$

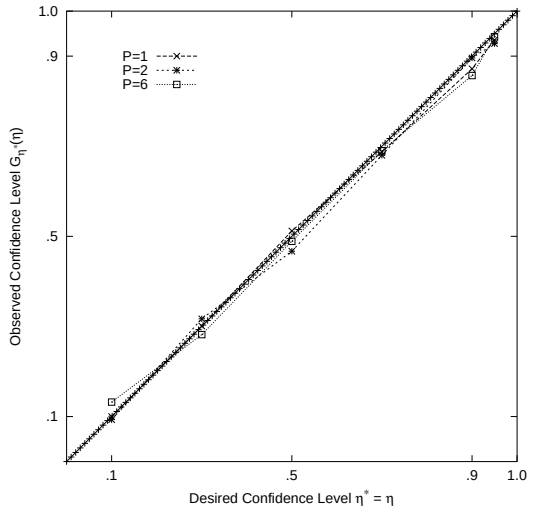


(a) NOBM

(b) SBM

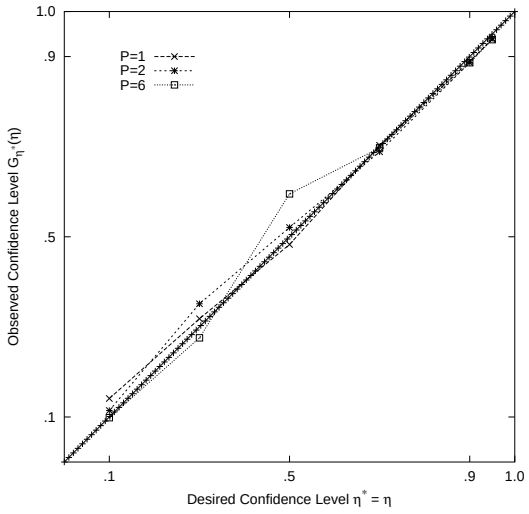


(c) OBM

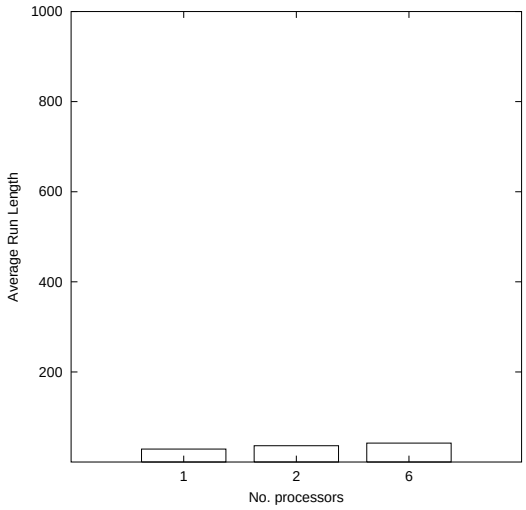


(d) STS/CSUM

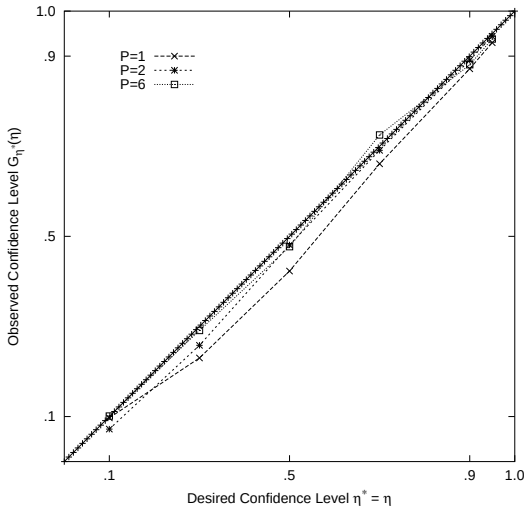
Fig. 11: Coverage function for mediumficient of variation and $\rho = 0.50$.



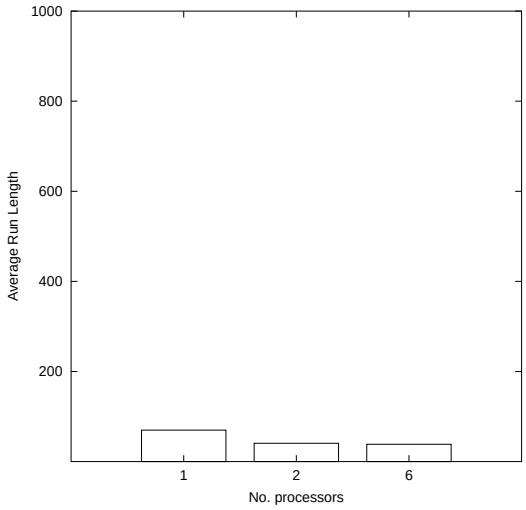
(a) NOBM coverage function



(b) NOBM run length



(c) CSUM coverage function



(d) CSUM run length

Fig. 12: Coverage function for medium coefficient of variation and $\rho = 0.50$.

the coefficient of variation of the service mechanism, imposes greater difficulties to methods of analysis to estimate the sample variance that is used to derive the confidence interval. As as result, coverage degrade, and the necessity of a robust methodology becomes stronger. Figure 13 illustrates this fact.

Note that by using OBM to analyze the output sequence of the M/D/1 simulation, as more processors are added the coverage improves, but not so close to the nominal confidence level. As C_x increases, we can perceive that coverage gets definitively closer to the nominal confidence level. Therefore, OBM performs much better under MRIP as the correlation structure becomes more complex. We are going to see in the next sections that this claim is also true for very high loaded systems. But this not occur for any CIP, just for the robust ones. For example, Figure 14 makes it clear that SBM does not always guarantee the quality of the results. As C_x increases, SBM seems not to find a good solution for the strong correlation and becomes wasteful (M/M/1) or insufficient ($M/H_2/1$).

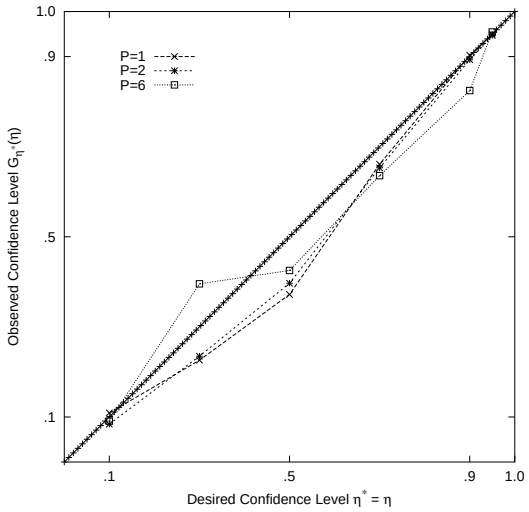
For all variants of *batch means*, the variability of confidence interval had identical behavior, that is, CoV{H} is around 0.20 for low values of confidence level, and decreased to 0.05 when the confidence level increased. We have simulated all CIPs using traffic load in the range $[0.10 \leq \rho \leq 0.95]$. Our results makes it clear that all proposed CIPs are asymptotic valid as they decrease in direction to zero as the run length increases. Figure 15 presents a comparison of this MOE for the proposed CIPs when simulating a 90%-loaded M/M/1 system.

Worthwhile to emphasize that by varying the degree of parallelism is especially helpful when the the variability is high. For all CIPs here investigated this occur when the confidence level is in the range $[0.10 \leq \eta \leq 0.50]$. Let us look at an example of this behavior in Figure 15. It summarizes the simulation of an M/M/1 queue with traffic intensity 0.90.

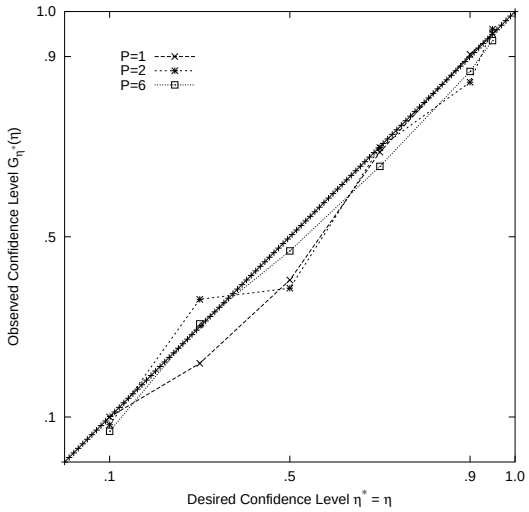
For confidence level in the region $[0.10 \leq \eta \leq 0.50]$, increasing the number of processors taking part in the parallel simulation always reduced CoV{H}. This behavior also occurred for higher traffic intensities (see the complete results of our experiments from Appendixes D to H.

With the methods based on STS this behavior is also true, but there is a particularity we would like to emphasize. Looking at Figure 16, we perceive that STS/CUM presented very high variability of confidence interval (around 0.40) for a single processor when the confidence level was in the range $[0.10 \leq \eta \leq 0.50]$, but even so this variability decreased as we added more processors.

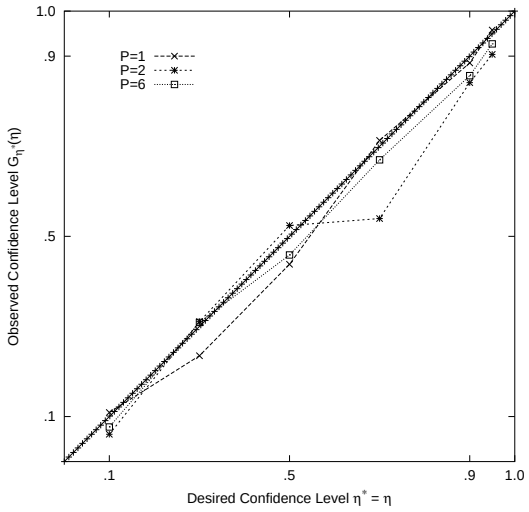
It becomes clear that increasing the degree of parallelization reduces CoV{H} whatever the traffic intensity and confidence level. These results confirm our claim that MRIP is (by itself) a natural variance reductor, no matter which method of analysis is being used. Section 5.8 will support this claim as it will also show that this feature of MRIP can be useful even when the method of analysis, though robust, is not properly configured.



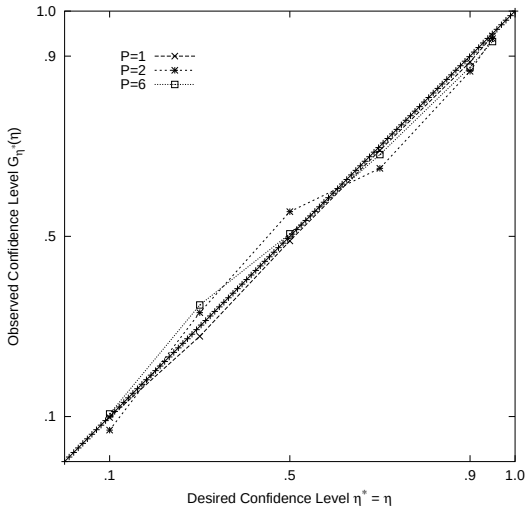
(a) $M/D/1$



(b) $M/E_4/1$

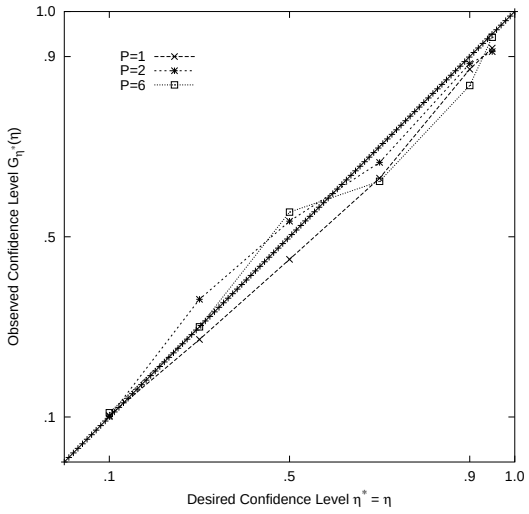


(c) $M/M/1$

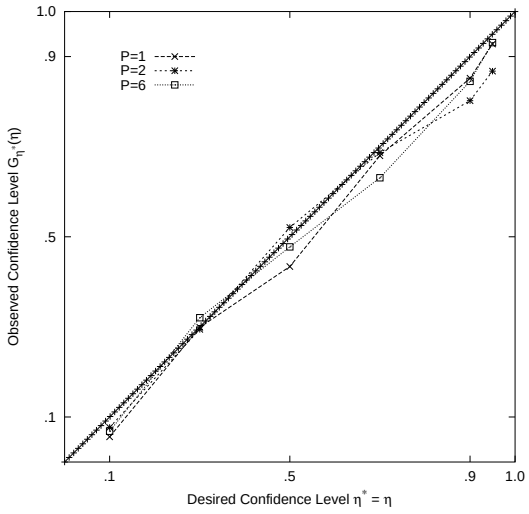


(d) $M/H_2/1$

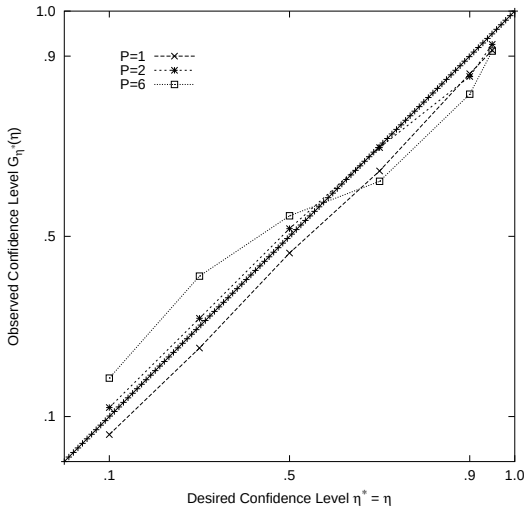
Fig. 13: Coverage function for OBM, $\rho = 0.90$ and increasing C_x.



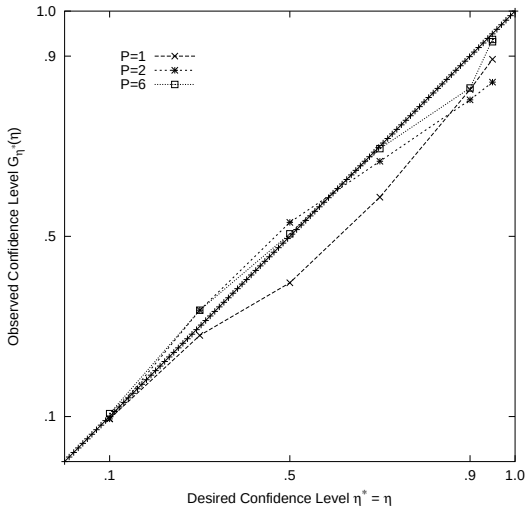
(a) $M/D/1$



(b) $M/E_4/1$

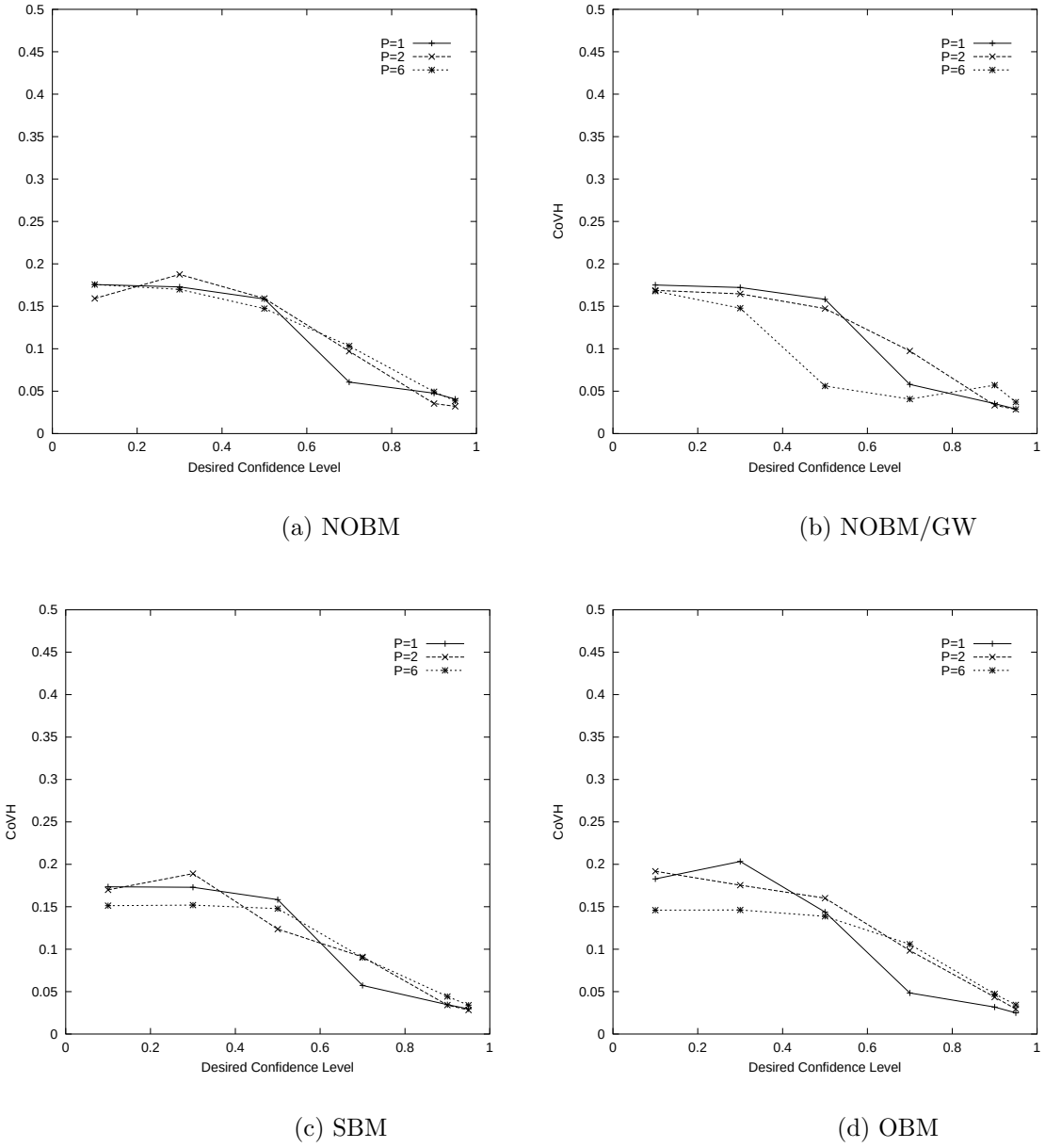


(c) $M/M/1$



(d) $M/H_2/1$

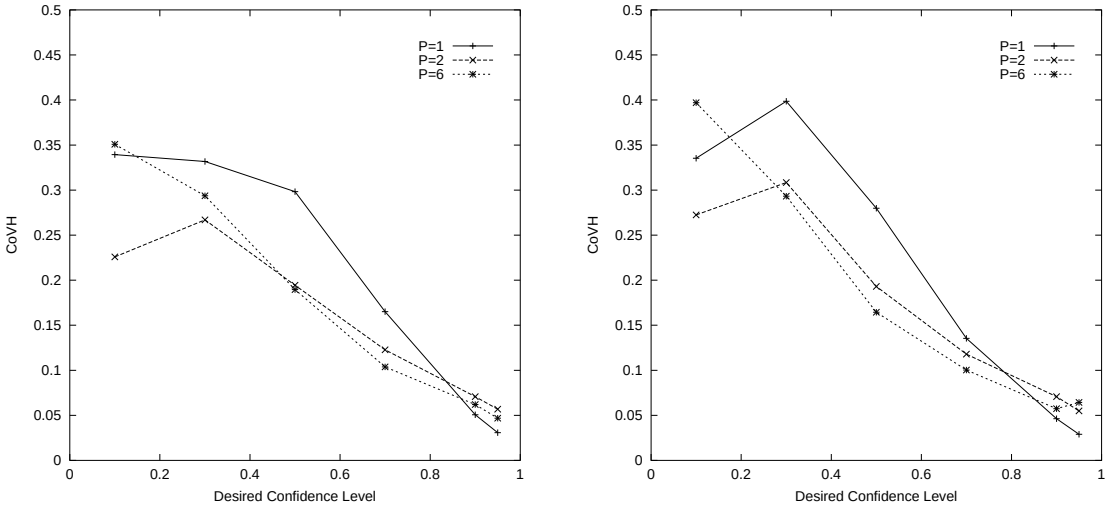
Fig. 14: Coverage function for SBM, $\rho = 0.90$ and increasing C_x.



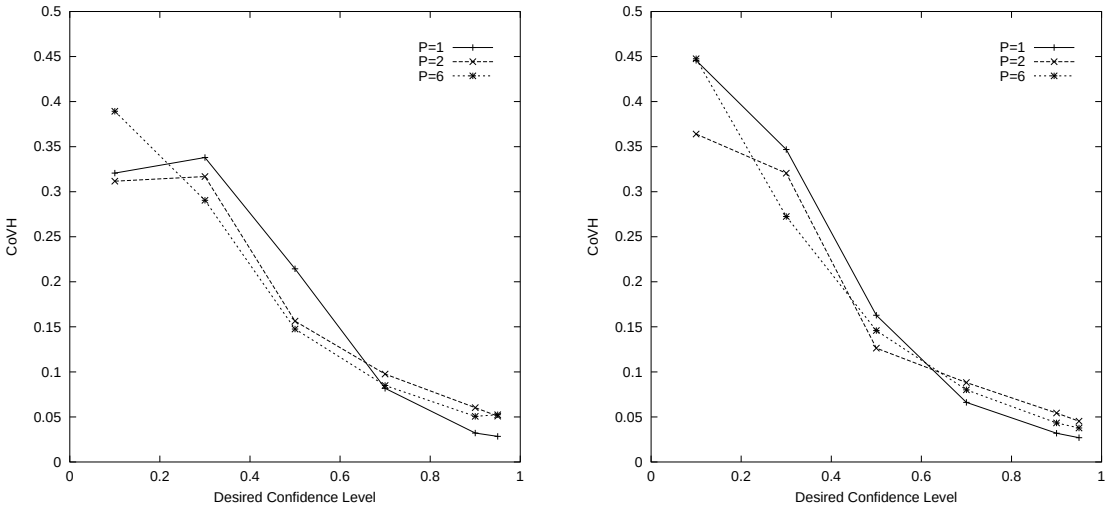
(a) NOBM (b) NOBM/GW

(c) SBM (d) OBM

Fig. 15: Cov{H} of CIPs based on batch-means-based : M/M/1, $\rho = 0.90$.

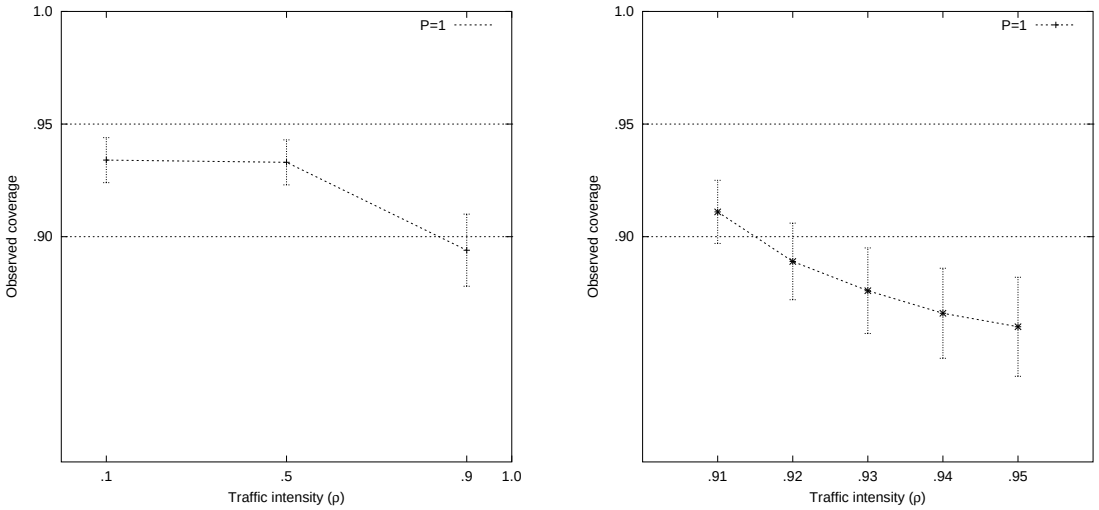


(a) $M/D/1$ (b) $M/E_4//1$



(c) $M/M/1$ (d) $M/H_2/1$

Fig. 16: CoV{H} of STS/CSUM for increasing C_x, and $\rho = 0.90$.



(a) $\rho \leq 90\%$ (b) $\rho > 90\%$

Fig. 17: NOBM performance for different ranges of ρ.

5.3 CIPs based on NOBM

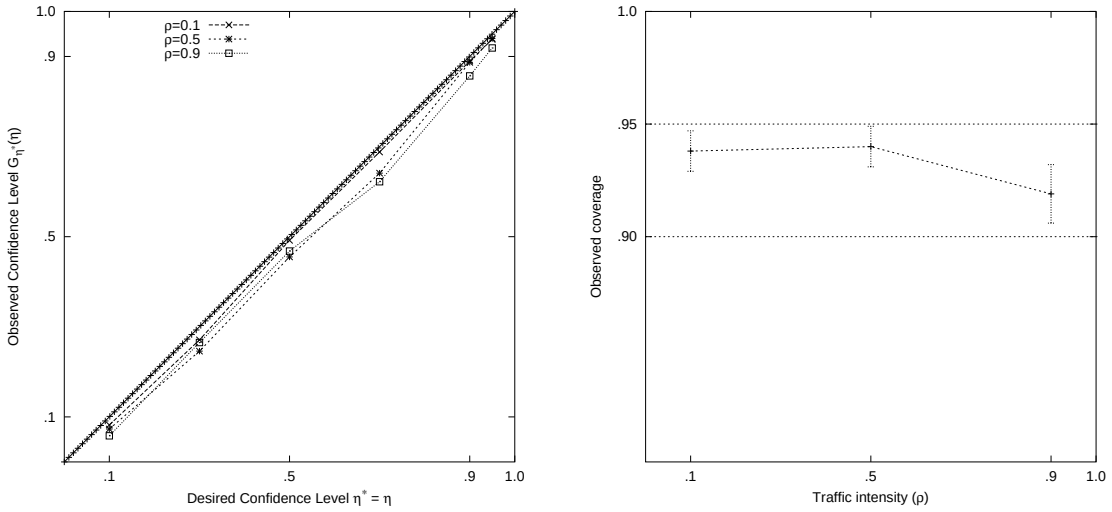
5.3.1 NOBM performance

Our experimental investigation showed us that NOBM behaves suitably for queuing systems moderately loaded. Figure 17a depicts the coverage function of NOBM for an M/M/1 under increasing traffic intensity ρ. The observed coverage gets close to the expected one, though for higher values of traffic one can perceive a trend to coverage erosion.

This trend has already been advised as we commented Bartlett's equation Equation 2.6, since as the simulated processes become more correlated, it turns out to be difficult to estimate the sample mean variance.

For highly loaded system (we consider high load the traffic intensity greater than 0.9), sequential coverage analysis captured the NOBM performance degradation, shown in Fig. 17b . By adopting a 5%-tolerance to the deviation of of the desired coverage, we can say that NOBM departs unacceptably from the zone of acceptance as the traffic intensity of the queuing system increases.

It does not seem reasonable to add more processors in order to improve the coverage of a sequential CIP, if it does not behave properly with one processor. This way, we investigated alternative CIPs to NOBM, trying to assess if they can perform suitably for very highly loaded systems. NOBM performance analysis for moderately loaded systems and different number of processors and confidence levels can be found in Appendix D.



(a) $\rho \leq 90\%$ (b) $\rho > 90\%$

Fig. 18: NOBM/GW performance for $\rho \leq 90\%$

5.3.2 NOBM/GW performance

As stated in the formulation of the proposal for this CIP, increase the number of batches is necessary in order to guarantee asymptotic validity. By estimating sequentially the average waiting time, the run length increases when (i) traffic intensity increases, as it means that correlation structure is stronger; (ii) requirements for stopping simulation are tighter, e.g. larger values of confidence level, or smaller relative precision; and (iii) service mechanism is more intricate (higher C_x). In all experiments we conducted, relative precision was not changed.

Figure 18 shows how the coverage of NOBM/GW behaves for increasing values of confidence level and traffic intensity. As expected, NOBM/GW is not an improvement on NOBM when run length is low/medium, but it put the final coverage inside the region of acceptance for ρ close to 90%. Thus, it deserves investigation for higher values of ρ.

For sake of comparison, we plot its results together with NOBM. Figure 19 makes it clear that NOBM/GW is an improvement on NOBM when run length is high, but does not produces coverage inside the acceptance region we have chosen. What is the price for this little improvement ? We must look at other MOE to assess this information.

Table 11 shows us that the computational effort (given by the average runlength over 1203 replications) to achieve this little improvement was not high, but there was reduction in the variability of confidence interval half-width.

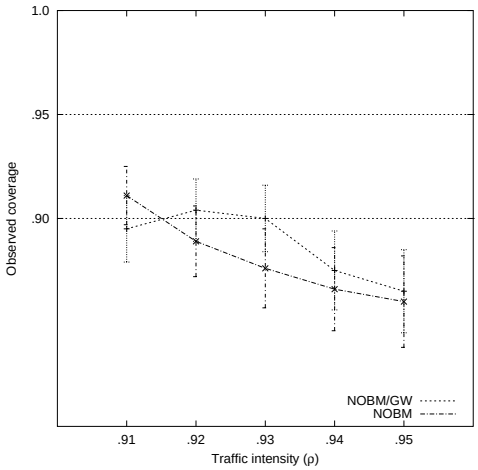


Fig. 19: NOBM/GW performance for $\rho > 90\%$

CIP	ρ	η	run length	CoV{H}	R
NOBM	0.95	0.95	1474.85	0.0382	1203
NOBM/GW	0.95	0.95	1501.72	0.0339	1065

Tab. 11: NOBM/GW : average run length and CoV{H} for $\rho = 0.95$ and $\eta = 0.95$.

5.4 SBM performance

This CIP takes advantage of the previous result, and reinforces the attack against correlation, which together could lead us to wait for better results. Figure 20 makes it clear that SBM with fixed spacing can be wasteful, since in the coverage function we perceive a trend to overestimate the confidence interval. On the other hand, it is a clear improvement on NOBM/GW for high values of traffic intensity, though it still doe not yield coverage inside the acceptance region. The price for that ? Let us check the other MOE.

Table 12 let us to conclude that the little improvement on NOBM/GW is worthwhile, as it requires not so much more observations, and the final confidence intervals are more stable.

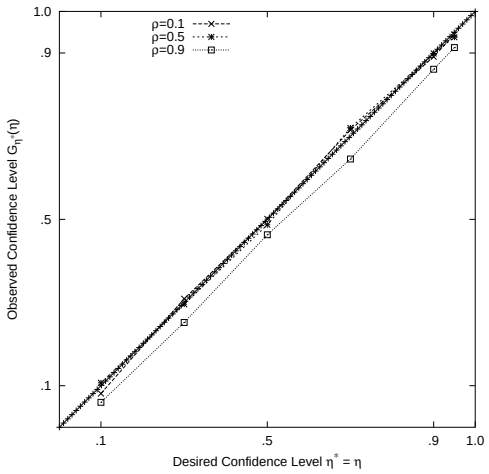
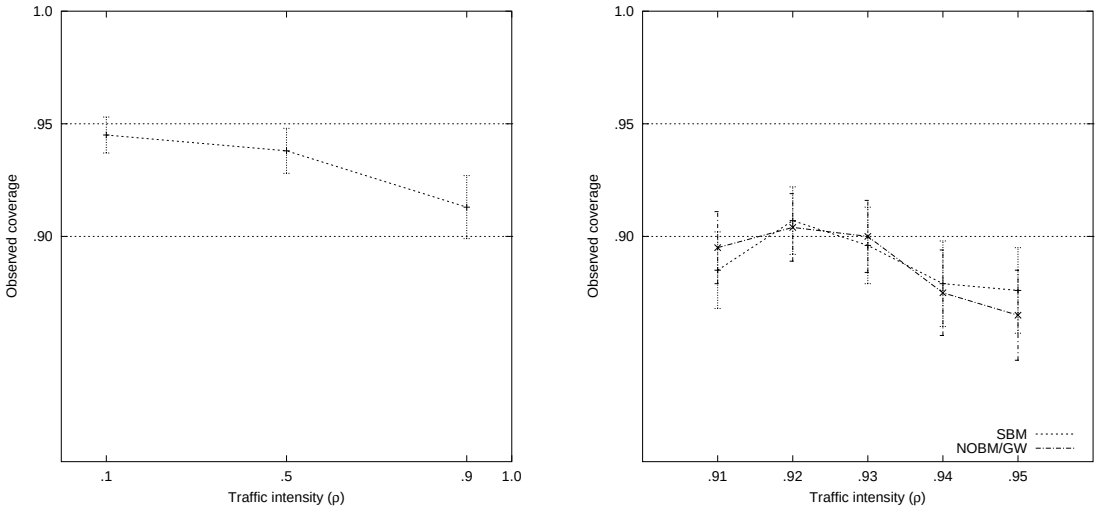


Fig. 20: SBM performance for $\rho \leq 90\%$



(a) $\rho \leq 90\%$ (b) $\rho > 90\%$

Fig. 21: SBM performance for different ranges of ρ.

CIP	ρ	η	run length	CoV{H}	R
NOBM	0.95	0.95	1474.85	0.0382	1203
NOBM/GW	0.95	0.95	1501.72	0.0339	1065
SBM	0.95	0.95	1505.77	0.0335	1243

Tab. 12: SBM : average run length and CoV{H} for $\rho = 0.95$ and $\eta = 0.95$.

5.5 OBM performance

We avoided using the results from NOBM/GW, successfully applied in SBM, since the increment of the number of batches is inherent to the method of OBM. Figure 22 shows that OBM can be a little wasteful when run length is low and we increase the degree of parallelization, but is behaves even better for higher traffic intensities. A look at Figure 23 makes it clear that OBM overcomes the previous CIPs in the sense of coverage of the results.

Table 13 tell us that to achieve so impressive performance, OBM collects around 1/3 more observations than SBM, making the variability of the confidence intervals much lower. This way, we have found the first candidate for validating our hypothesis H_5, namely, we can generate in parallel the amount of observations OBM requires, making it not only accurate, but also fast.

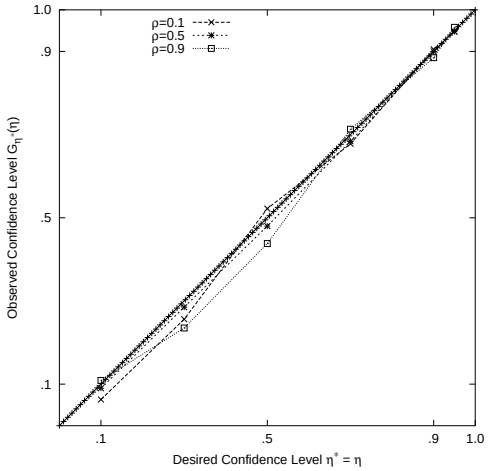
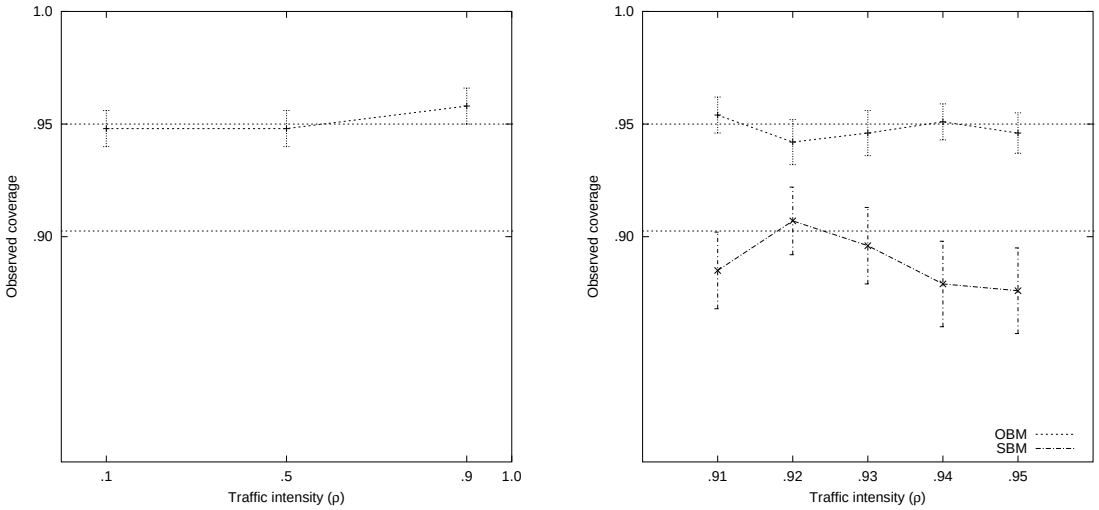


Fig. 22: OBM performance for $\rho \leq 90\%$

CIP	ρ	η	run length	CoV{H}	R
NOBM	0.95	0.95	1474.85	0.0382	1203
NOBM/GW	0.95	0.95	1501.72	0.0339	1065
SBM	0.95	0.95	1505.77	0.0335	1243
OBM	0.95	0.95	2214.62	0.0258	2333

Tab. 13: OBM : average run length and CoV{H} for $\rho = 0.95$ and $\eta = 0.95$.



(a) $\rho \leq 90\%$ (b) $\rho > 90\%$

Fig. 23: SBM performance for different ranges of ρ.

5.6 CSUM.1 performance

Figure 24 depicts that when the confidence level is above 0.70, CSUM.1 yields always coverages very close to the nominal confidence level, specially when we add more processors under MRIP. Compared to OBM for traffic intensity above 0.90 (see Figure 25b), CSUM.1 yields somewhat less accurate result, but always inside the acceptance region we have defined earlier.

Table 14 shows two undesirable properties : CSUM.1 requires more observations than other CIPs here analyzed, and even so can not reduce the variability of the final confidence interval. These disadvantages can be compensated with higher degree of parallelization, as we shall see in the next section.

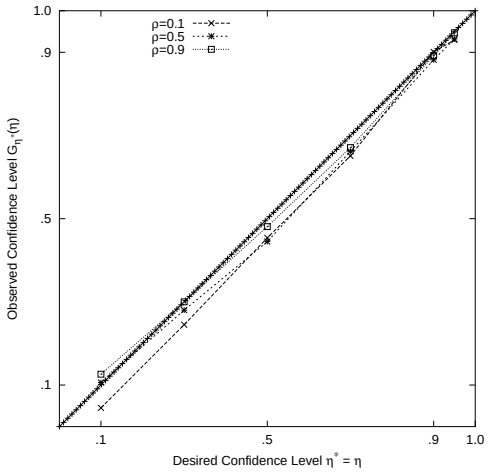
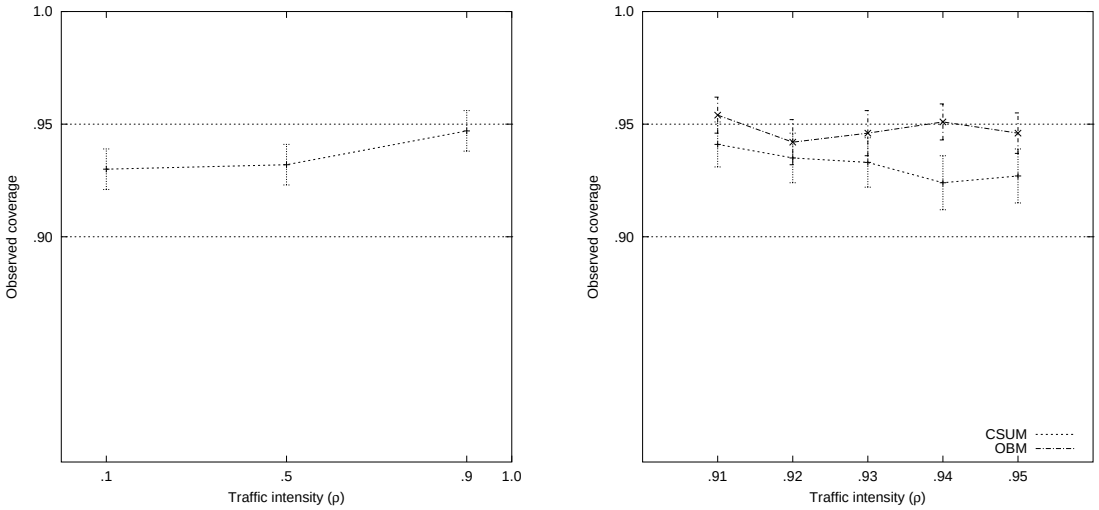


Fig. 24: CSUM.1 performance for $\rho \leq 90\%$



(a) $\rho \leq 90\%$ (b) $\rho > 90\%$

Fig. 25: CSUM.1 performance for different ranges of ρ.

CIP	ρ	η	run length	CoV{H}	R
NOBM	0.95	0.95	1474.85	0.0382	1203
NOBM/GW	0.95	0.95	1501.72	0.0339	1065
SBM	0.95	0.95	1505.77	0.0335	1243
OBM	0.95	0.95	2214.62	0.0258	2333
CSUM.1	0.95	0.95	2239.59	0.0492	2016

Tab. 14: CSUM.1 : average run length and CoV{H} for $\rho = 0.95$ and $\eta = 0.95$.

5.7 Performance under very high traffic intensity

By adding more processors in an efficient environment such that provided by Akaroa-2, we are certain that observations will be generated more quickly, besides being of better quality, since they come from independent simulation engines, carefully initiated with nonoverlapped pseudorandom numbers.

Table 15 shows that OBM collects almost the same amount of observations, as we increased the degree of parallelization. Moreover, it remains no doubt that OBM is the most accurate CIP for high traffic intensity.

CSUM.1, on the other hand, continues yielding acceptable coverage, though lower than OBM (see Table 14), and besides producing more quickly the observations under MRIP, CSUM.1 performed suitably for operating in this environment. Both set of experiments validated our hypothesis H_5.

OBM	ρ	η	run length	CoV{H}	R
P=1	0.95	0.95	2214.62	0.0258	2333
P=2	0.95	0.95	2230.46	0.0260	1899
P=4	0.95	0.95	2220.21	0.0262	2306
P=6	0.95	0.95	2143.38	0.0297	2134

Tab. 15: OBM : average run length and CoV{H} for higher degree of parallelization

CSUM.1	ρ	η	run length	CoV{H}	R
P=1	0.95	0.95	2239.59	0.0492	2016
P=2	0.95	0.95	1915.31	0.0291	1644
P=4	0.95	0.95	1811.59	0.0305	1709
P=6	0.95	0.95	1695.03	0.0301	1490

Tab. 16: CSUM.1 : average run length and CoV{H} for higher degree of parallelization

5.8 Compensating imprecise batch size selection

5.8.1 OBM

We look for manners to degrade the performance OBM, in order to validate our hypothesis H_2, which states that imprecise batch size selection can be compensated with higher degree of parallelization. Reasons for that hypothesis is twofold : (i) OBM can generate a large amount of batches; and (ii) observations generated in MRIP have better statistical quality than those arising from a single replication.

In this experiment, we configured OBM to use only 40 batches for generating correlation coefficients by means of jackknife method. Jackknife estimators of correlation coefficients are less unbiased, as long as 100 batches are used. As we simulated a 95%-loaded M/M/1 queue to construct a confidence interval at 95% of confidence level, the coverage for P=1 was 0.879, outside the zone of acceptance (see Figure 26. As long as we added more processors, coverage could be improved.

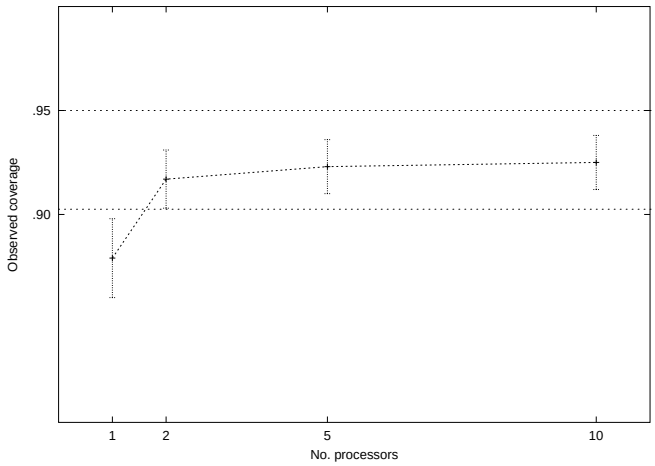


Fig. 26: OBM : compensating imprecision of batch size selection

5.8.2 CSUM.2

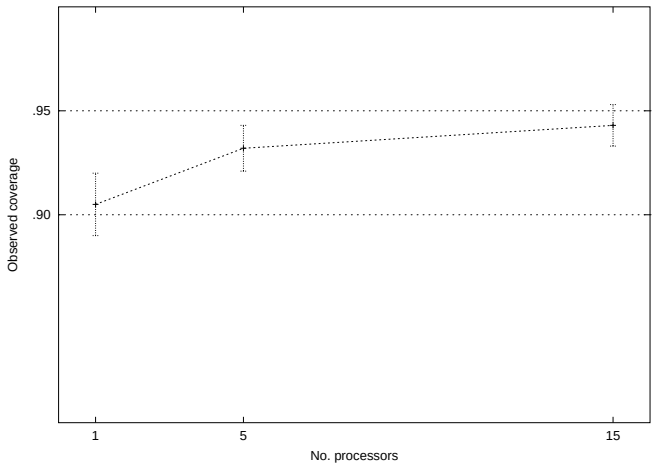


Fig. 27: CSUM.2 : compensating imprecision of batch size selection

In this experiment, we have used the CSUM version with fixed number of batches (CSUM.2). We forced imprecision of selection of batch size by using a lower level of significance for the test of normality. For normal data the Shapiro-Wilk statistic W should be near 1. Since we have adopted a number of batches equal 10, the level of significance closer to 1 should be greater than 0.95 (see Appendix A). With a level of significance 0.95, CSUM.2 yields good performance. The use of higher level of significance denotes the necessity of larger sample size, and the corresponding improvement is very little.

On the contrary, by choosing a lower level of significance, test for normality succeeds quicker, and the selected batch size is not the optimal, reducing the computational demand of this phase. In this experiment we used a level of significance 0.01.

By simulating a 95%-loaded M/M/1 queue to construct a confidence interval at 95% level, coverage using a single processor was worst than for the previous scenario that uses higher level of significance. To compensate this negative effect, we increased the degree of parallelism, and the coverage could be improved, getting closer to the nominal confidence level.

Table 17 shows that for P=1 CSUM.2 does not collect sufficient data for that combination of ρ and η. Look at Table 14 that CSUM and OBM must collect over $2.2x10^6$ observations in order to produce accurate coverage. As we add more processors to work together, CSUM.2 yields good coverage with less observations (around $2.1x10^6$ observations).

Another desired effect is over the variability of confidence intervals that decreased as we added more processors. The large number R of times we repeated the experiment (exclusive too short executions), signalizes that we did not take it by chance.

$CSUM.2$	ρ	η	run length	CoV{H}	R
P=1	0.95	0.95	1870.73	0.0366	1544
P=5	0.95	0.95	2171.84	0.0285	1964
P=15	0.95	0.95	2111.96	0.0272	2027

Tab. 17: CSUM.2 : average run length and CoV{H} for higher degree of parallelization

5.9 Speedup issues

According to Amdahl's law [4], if a fraction f of a computation is inherently sequential, then the speedup S(P) is bounded above by

$$S(P) = \frac{1}{f + \frac{1-f}{P}} \tag{5.1}$$

where P is the number of processors and f is defined to be the ratio of the service demand of sequential parts of the computation to the service demand of the entire computation.

In steady-state simulation, considering that results are analyzed sequentially, MRIP imposes a limit to the average speedup that should be incorporated into above expression.

Let N_{min} be the number of collected observations, sufficient for achieving the required precision of the final results. One could think of a situation in which there are so many processors employed that each one achieves just the first checkpoint, and the stopping rule is reached. Let this number of processors be P_{min}.

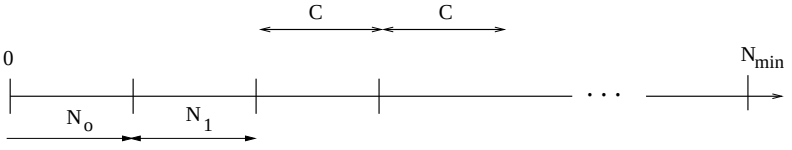


Fig. 28: Steady-state structure

Let N_o be the length of the transient phase, N_1 be the amount of steady-state observations from the end of the transient phase until the optimal batch size is found. Our implementations consider that the first checkpoint occurs at this moment when the optimal batch size is determined. Let C be the distance between consecutive checkpoints (Figure 28).

A truncated version of Amdahl's law for the MRIP scenario, formulated by Pawlikowski and McNickle [102] states that the speedup achievable with P homogeneous simulation processors is given by :

$$S(P) = \begin{cases} \frac{1}{f+\frac{1-f}{P}} & for \quad P \leq P_{min} = \frac{(1-f)N_{min}}{C} \\ \\ \frac{1}{f+\frac{1-f}{P_{min}}} & for \quad P > P_{min} = \frac{(1-f)N_{min}}{C} \end{cases} \tag{5.2}$$

The speedup obtained from the truncated Amdahl's law of Equation 5.2, when assuming $N_{min} = 10000$ and C=1, is plotted in Figure 29. One can see that linear speedup would only be possible if f=0. This occurs when the simulation engines work cooperatively in parallel. As the value of parameter f increases, the speedup falls away from the theoretical trajectory.

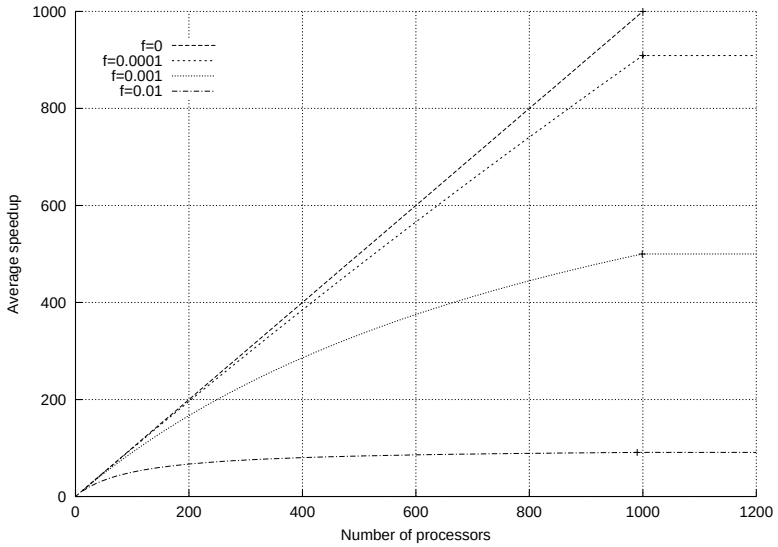


Fig. 29: Speedup achievable theoretically under the MRIP scenario according to the truncated Amdahl's law ($N_{min} = 10000$)

Observations collected during initial transient phase (N_o) are not processed in parallel, therefore observations collected during this period belong to f. Thus,

$$f = \frac{N_o}{N_{min}}$$

Note that if C means the number of observations collected between consecutive checkpoints, then P_{min} times C observations are needed to stop the simulation. That is,

$$P_{min} = \frac{(1-f)N_{min}}{C} \tag{5.3}$$

When using P_{min}, one achieves the maximum speedup given by

$$S_{max} = \frac{N_{min}}{fN_{min} + C} \tag{5.4}$$

To assess the average speedup obtained when the sequential batching techniques considered are applied in *Multiple Replications in Parallel* scenario, we simulated an $M/M/1$ queuing system, with traffic intensity 95%, and constructed a confidence interval at 95% of confidence level. The stopping rule was, as in the every experiment of this research, a relative precision of 5%.

A specific feature of sequential techniques is that randomness of data collected at the output of the model being analyzed can fortuitously yield the stopping condition much earlier than it could be expected, and that can lead to wrong results. In light of that, we adopted the practical methodology proposed by Ruth Lee et al. [87]. Namely, while using P processors we:

1. run the simulation experiment 3 times;
2. accepted the results produced by the longest simulation run only;
3. recorded the average length of the transient phase, measured by the number of transient observations $\bar{N}_o$ discarded by each of P processors;
4. recorded the average number of observations $\bar{N}_1$ required to achieve the first checkpoint;
5. recorded the average total number of observations $\bar{N}_{min}$ when simulation was stopped.

Each time when these steps were followed, we obtained an n-uple $(\bar{N}_o, \bar{N}_1, \bar{N}_{min})$. The average fraction $\bar{f}$ of the simulation which cannot be parallelized is calculated automatically, that is,

$$\bar{f} \;=\; \frac{\bar{N}_o}{\bar{N}_{min}} \tag{5.5}$$

To improve the accuracy of the results we repeated the above sequence 100 times and averaged the results at the end. We repeated the whole experiment for P=1, 5, 10, 15 and 20 processors. P_{min}, the number of processors that could still give a speedup, was calculated from the truncated Amdahl's law, using the results obtained from simulations on P=1 processor.

Considering the classical NOBM, NOBM/GW, SBM, and CSUM.1, C is at least equal to m^*, and the procedure has little flexibility for tuning C. OBM offers greater flexibility as C can be smaller than m^*. Namely, in the case of complete overlapping, the distance between checkpoints can be, theoretically, as short as 1. Thus, for f remaining the same as in the previous methods, P_{min} can be considerably greater in OBM.

CSUM.2 offers relative intermediate size of C, since the procedure can collect a number of observations as small as a multiple of b (instead of m^*), and, since $b = 10$, we could achieve a reasonable checkpoint granularity.

Worthwhile to warn at this point, that the extreme cases for OBM and CSUM.2 (checkpoints at a distance of 1 and 10 observations, respectively), were tried but each replication took extremely very long time (around 6 days), and we should abort them. Probably, the reason for that can be the same given by Pawlikowski and McNickle [103] : "the global analyzers of Akaroa-2 have not been designed to maximize speedup."

In order to achieve sound speedup, we should decrease the extension of batch size selection phase. As already stated, by estimating directly the correlation coefficients by means of jackknife method is a trial to improve the quality of the result, but it has two disadvantages :

1. It is time-consuming for long run length; and
2. It requires at least 100 batches to yield accurate estimates.

The test of independence based on the von Neuman statistic, according to Kleijnen [76], also requires at least 100 batches to yield reliable results, though Fishman [39], Alexopoulos [2], Steiger and Wilson [125] agree it can be as small as 8 batches.

We implemented the rank version of von Neuman's test, since it is less time-consuming, and plugged it in our sequential version of OBM. We simulated the mean waiting time of clients of an M/M/1 queuing system, 95% loaded, to construct a 95%-confidence interval. Checkpoints occur whenever a new batch is collected, that is, $C = m^*$. By using $b = 10$ and $b = 20$, for increasing degree of parallelization, we obtained the following results :

P	ρ	η	cov b = 10	cov b = 20	run length b = 10	run length b = 20
1	0.95	0.95	0.591	0.699	465.289	725.858
5	0.95	0.95	0.647	0.756	570.442	878.491
10	0.95	0.95	0.701	0.774	598.199	924.549
15	0.95	0.95	0.714	0.785	602.787	930.406
20	0.95	0.95	0.689	0.806	611.691	963.221

Tab. 18: OBM : von Neuman test of independence

It is clear that small number of batches yields very poor coverage, though it can be improved by increasing the degree of parallelization. Using more batches improves a little the coverage, but once again more processors under MRIP increases the accuracy of the coverage. Increase of run length can be accepted as a consequence. But how much processors can we apply when adopting $b = 10$?

By applying Equation 5.3 for this experiment, we have found that the maximum number of processors that can be work together is 1215. Let us see which additional information the truncated Amdahl's law gives to us.

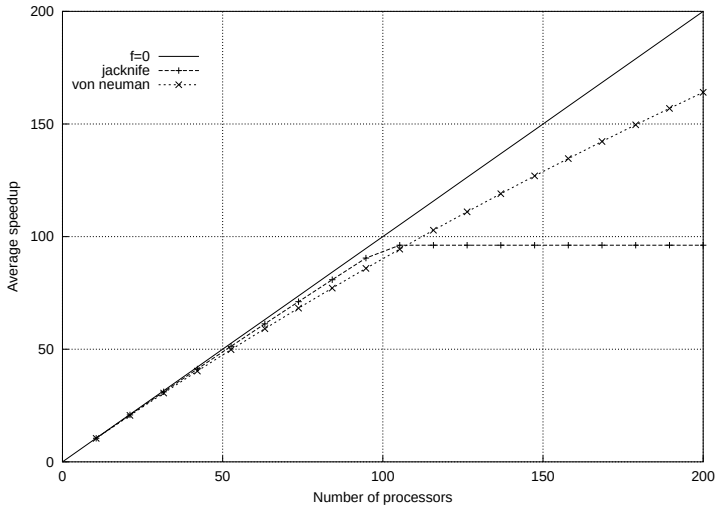


Fig. 30: OBM speedup : checkpoint at every m^* observations

Figure 30 depicts the sound speedup OBM can give us in this scenario, when compared to the OBM version using jackknife estimators of correlation coefficients to assess the independence among batch means. By running both experiments with increasing degree of parallelization (P=1,5,10,15,20), we found f = 0.0005 for OBM with jackknife estimators, and f = 0.0011 for OBM with the test of independence based on the von Neuman. Practically the same, but the former needs at least 100 batches to find the optimal batch size, while the latter used only 10 batches. As expected, batch size selection phase was reduced.

By reducing the phase of choosing the batch size, we automatically reduced the distance between consecutive checkpoints, since in this implementation $C = m^*$, and at the same time the knee described by Amdahl's law was shifted, meaning that we could apply more processors.

On the basis of truncated Amdahl's law 5.2, the above configuration of the sequential version of OBM using von Neuman test, reduced both f and C, which corresponds to an increase of P_{min}, according to Equation5.3. Since P_{min} is the minimum number of processors to yield the maximum speedup, it corresponds to the knee on the graphic of the truncated Amdahl's law.

To complete our investigation on speedup of the two batching procedures that our research found to yield the best performance under MRIP, let us use the same example of Section 5.8.2, when we forced the imprecision of selection of batch size by using a lower level of significance ($\beta = 0.01$) for the test of normality.

Being faster (though imprecise), the test for normality will decrease the batch size selection phase. Figure 31 depicts what we hypothesized (H_2) : Being consistent and robust the method of analysis, MRIP can be of much further importance in the sense of speeding up good results.

By simulating a 95%-loaded M/M/1 queuing system to construct a confidence interval at 95% level, we found f = 0.0003 when using $\beta = 0.95$, and f = 0.0004 when using $\beta = 0.01$. Difference appeared in the reduction of batch size selection for the latter case, allowing that up to 78 processors can cooperate together to accelerate the experiment, agains 25 for the other case.

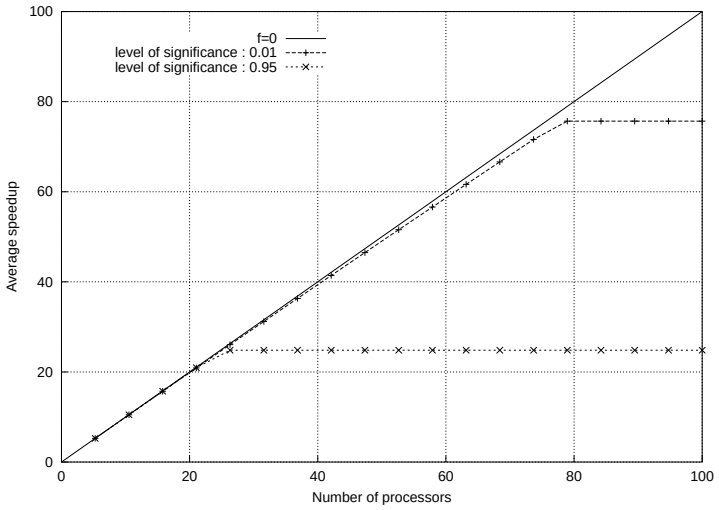


Fig. 31: CSUM.2 : checkpoint at every m^* observations

6

Case study

6.1 Introduction

In the previous chapter we stressed the design issues of some sequential procedures based on *Batch Means* and *Standardized Time Series*. In this chapter we are going to choose one of those procedures in order to use it for automatically estimating steady-state mean of a performance parameter of a wireless communication system.

Since any statistic from such experiments cannot be guaranteed to give a close estimate for every sample, we must design statistics that will give acceptable results on the average or in the long run. By using a well designed sequential analysis procedure for analyzing data carefully, we can guarantee a better quality of the conclusions drawn from the analysis.

6.2 Problem statement

The practical problem of interest (PPI, according to the adopted methodology depicted in Fig. 32) was proposed by Fitzek and Wolisz [42]. It is an adaptation of the "Simultaneous MAC Packet Transmission (SMPT)" approach to a CDMA based mobile communication system with a varying number of mobile users.

To support the increasing demand for highly heterogenous QoS requirements by the mobile users, the authors proposed the statistical multiplexing of CDMA channels. SMPT attempts to stabilize the Quality Of Service (QoS) of a wireless CDMA system in terms of throughput, loss rate and delay even if the propagation conditions on the wireless medium change dramatically.

Using the simplest ARQ mechanism *Send and Wait*, like it is discussed in [12] and suggested within the recent wireless LAN standards (IEEE 802.11 and HiperLAN1), each erroneous packet is retransmitted while following stored packets have to wait until the packet has been transmitted successfully.

The effective bit rate decreases from B_{good} to B_{bad} and simultaneously the jitter increases. We assume that a resulting bit rate B_{bad} is not acceptable for the required throughput specified by the QoS parameters. Further the increased jitter is not acceptable for the application.

We consider a CDMA based mobile communication system with a specific number of **W**ireless **T**erminals *(WTs)*. All WTs communicate with one central **B**ase **S**tation *(BS)*, which coverage defines the cell boundaries (see figure 32).

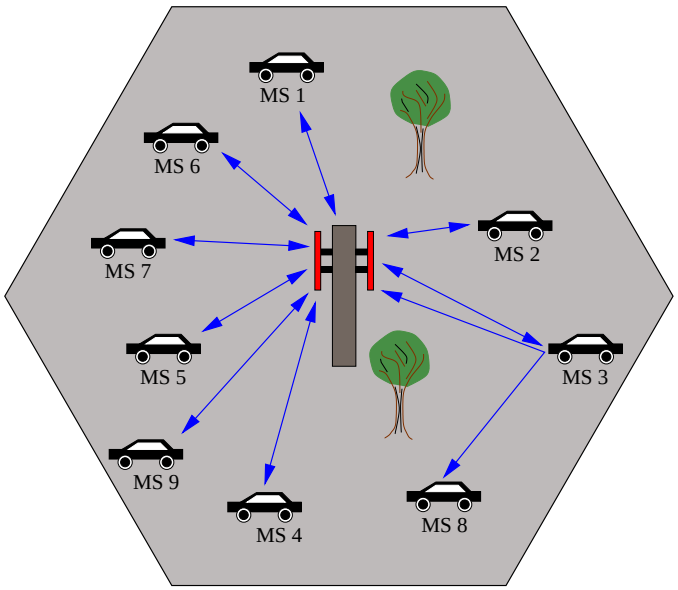


Fig. 32: Nine wireless terminals communicating with one base station

The mobile communication system supports a number of codes much higher than the number of active WTs. All WTs are sending asynchronously as well on bit level as on chip level. The wireless link is considered to be unreliable with a varying ***B**it **E**rror **P**robability (BEP)*. The value of the BEP depends on the number of active channels k.

For the chosen scenario we assume an **A**dditive **W**hite **G**aussian **N**oise (AWGN) channel with **B**ineary **P**hase **S**hift **K**eying (BPSK). Codes will be assigned before the connection is established. The total number of codes per mobile is set by the QoS requirements of the mobile.

We have performed a simulation study using the `Ptolemy` simulation tool [107] and for full parallelization, statistical evaluation and run length control we have used Akaroa with the `Ptolemy` interface akstars (see 33). It is worthwhile to say, that no additional effort was required from the analyst, and we could say that the framework can be still considered transparent from the user point of view.

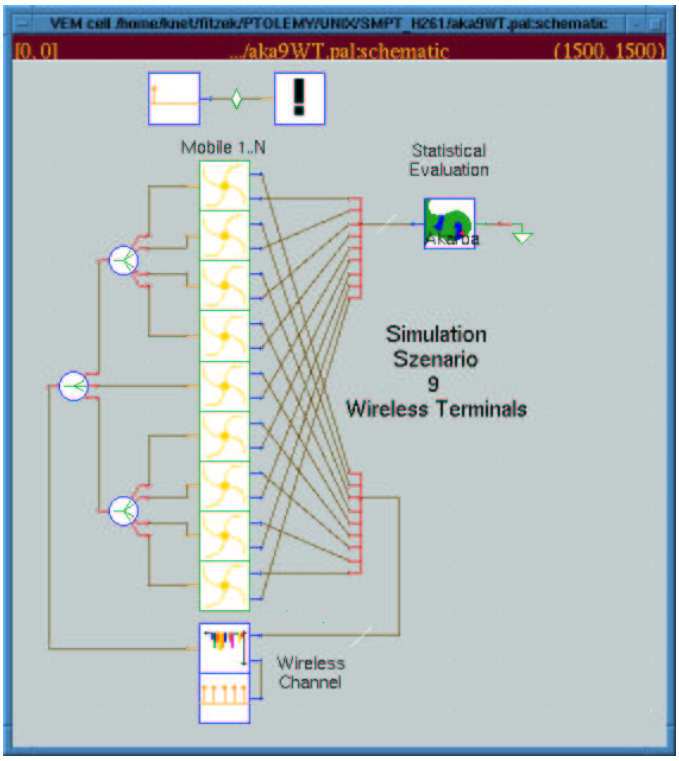


Fig. 33: Ptolemy structure

We formed a communication system with 9 WTs and one base station. Outer-Cell interference was not taken under consideration. The channel between the WT and the BS is modeled with a multilayered Markov chain, considering two channel states (*bad* and *good*) and the impact of used channels on the BEP on bit level. The main parts of the simulation model are the protocol implementation of the WT and BS.

Each WT generates a stream of transport units (like UDP segments and therefore called *segments*) with a specific load and pass these segments to the DLC layer via the network layer, where each of them is divided into a group of DLC packets. To each packet a header with length ζ is added. This header ζ is used to identify DLC packets in the right order and to assign the DLC packets to the appropriate segment and means for error detection. The frame, which is composed by one DLC packet and the header ζ is called a **D**ata **L**ink **C**ontrol **P**acket **D**ata **U**nit (DPDU).

The length of a DPDU is denoted as L_{DPDU}. All DPDUs are stored in an queue with a fixed length L_{Queue} within the DLC layer and will be sent with different ARQ based transmission methods over the wireless link. Packets that are prone to errors only by one bit error will not be decoded successfully on the receiver side and will be counted as loss.

The load generation module generates packets with constant bit rate allowing variable as well as fixed segment sizes with a specific load. We neglect the fading effects and assume an optimal power control within the WTs. Nevertheless using the wireless channel each WT will influence other WTs by an increased background noise.

6.3 Choice of a sequential method of analysis

We investigated the influenced jitter with high accuracy. Because of the nature of the wireless link, which is influenced by the user's mobility and the active mobiles using the wireless channel, is very difficult to predict how many observations values are necessary. Moreover, the simulation of the wireless channel on bit level is very time consuming. Therefore a fast simulation strategy is desirable.

Among those procedures investigated, we found that OBM and CSUM behave better when the experiment demands very long time to yield reasonable accurate results, OBM being the most accurate for high traffic intensity.

On the other hand, in [95], we showed that although (complete) OBM can be tuned to reduce the distance between consecutive checkpoints, the convergence rate to the desired relative precision becomes slower as the distance between consecutive checkpoints decreases. It means that OBM can offer good speedup, but one should always take into account an important MRIP property : The trade-off between fine granularity and achievable speedup.

CSUM presents shorter preprocessing task, namely that one for achieving normality among A_i's. This feature is very attractive, but that CIP needs some additional tuning to mitigating the higher variability of the final confidence interval. Therefore, we adopted the sequential version of complete OBM for this simulation study, since it fulfill the following criteria of classification (CCI, according to the adopted methodology depicted in Fig. 32) : it is more accurate, requires somewhat less observations than CSUM, and the achievable speedup is acceptable.

6.4 Simulation results

Simulation experiments were executed aiming to construct a confidence interval at 95% of confidence level, and the stopping rule adopted was a 5%-relative statistical error. To avoid using the results obtained by too short simulation runs we ran each experiment three times, using different sequences of pseudorandom numbers, and accept the results produced by the longest run.

Fig. 34 shows that for every scenario there is a reduction in time, as we divided the execution time by using one processor T(1), by the execution time by using 10 processors T(10). Results were obtained for 3,6, and 9 WTs, and two values of confidence level (95% and 99%). It shows a very important property of MRIP : it works even better as the run length increases.

Although we have used an almost homogeneous set of processors, the above result can be to some extent deceived, as it can vary with the system load. A better source of information would be the number of observations required to stop the simulation.

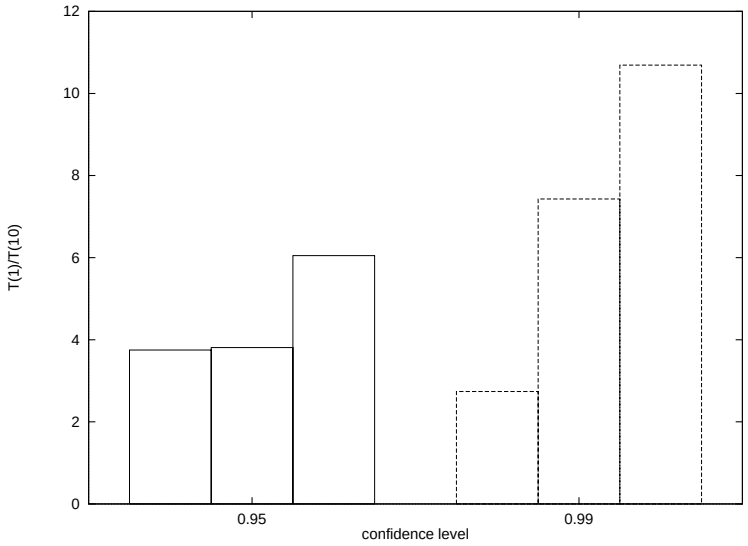


Fig. 34: Reduction of the simulation time of the jitter of a wireless link

WT	P = 1	P = 10
3	0.51 / 3645 / 243	0.51 / 2916 / 486
	0.52 / 3705 / 247	0.51 / 3108 / 518
	0.50 / 3885 / 259	0.52 / 2964 / 494
6	0.51 / 4464 / 248	0.54 / 2976 / 496
	0.51 / 4035 / 269	0.51 / 1614 / 269
	0.50 / 4608 / 256	0.51 / 3072 / 512
9	0.53 / 2988 / 249	0.53 / 4494 / 749
	0.53 / 3585 / 239	0.54 / 5736 / 956
	0.52 / 3660 / 244	0.52 / 4410 / 735

Tab. 19: Simulation results for 3,6 and 9 wireless terminals

Table 19 and Table 20 depict the performance results for the same number of wireless terminals as we have used 1 and 10 processors, for 95% and 99% confidence levels respectively.

For each wireless terminal of each scenario, the first value stands for the jitter estimate, the second value stands for the sample size required to achieve 5% of relative width, the adopted stopping rule, and the third value stands for the length of the transient phase, i.e. the number of discarded observations before achieving the steady state.

One can clear observe that when we use more processors to cooperate with independent observations, the framework based on MRIP is much more efficient as it needs less observations to achieve the required relative precision. The different (short) length of the transient phase justifies the application of the stationary test, instead of discarding an arbitrary number of observations, which would lead to statistical imprecision.

The experimental analysis showed us that the SMPT method provides better results in the sense of jitter. Before using Akaroa-2 together with OBM, the simulation experiment took around 188 hours, and it was clear if the statistical error could

WT	P = 1	P = 10
3	0.51 / 5103 / 243	0.51 / 7290 / 972
	0.52 / 5187 / 247	0.51 / 6072 / 765
	0.50 / 4662 / 259	0.51 / 7554 / 1000
6	0.51 / 8928 / 248	0.54 / 4464 / 744
	0.52 / 8877 / 269	0.51 / 3228 / 538
	0.50 / 9216 / 256	0.53 / 4572 / 762
9	0.53 / 10458 / 249	0.55 / 9777 / 486
	0.53 / 10038 / 239	0.53 / 13623 / 518
	0.52 / 10248 / 244	0.53 / 12820 / 494

Tab. 20: Simulation results for 3,6 and 9 wireless terminals

still be reduced, because due the very long time to yield an estimate, we were not stimulated to repeat the experiment with tighter requirements.

7

Summary, conclusions, and future research

7.1 Introduction

In the previous chapters, we stressed the necessity of designing confidence interval procedures for estimating the mean value of a stationary process. Moreover, despite the plethora of methodologies found in the literature, very few is known about the performance of sequential procedures based on these methodologies that can be run in an environment of multiple replications in parallel, a simple yet effective way of exploring the computing power of PCs connected via a network infrastructure.

In the sequence, we designed some sequential procedures based on the batching approach, and after adjusting them to get the best performance, we conducted experimental to validate our hypothesis, and investigated some issues aiming to get sound speedup with possibly less observations.

In this chapter, we summarize the results, draw some conclusions about the experience gained in this research, and point out directions for future research.

7.2 Summary

Application of computationally efficient CIPs in stochastic simulation of networks with multiple queues and servers is difficult but this is an important research topic. Another issue is the asymptotic behavior of these CIPs under MRIP, in order to anticipate their behavior as the run length increases.

Sequential procedures here presented, based on NOBM/GW and SBM, represent improvements over the classical NOBM, but they can not get rid of the problem

of strong correlations found in high traffic intensities and when we pass from an exponential to a hyperexponential service mechanism.

To cope with this scenario, sequential procedures based on OBM and STS/CSUM that can be run under MRIP were proposed, implemented and investigated. Indeed, they overcome the other two, in the sense that they yield accurate results, especially when traffic intensity is very high (e.g. $\rho > 0.90$). These results were found using just one processor.

Having detected the two more promising sequential procedures, we faced with the problem of collecting observations faster while the coverage of the results at a desired niveau. These characteristics suggest that a potential candidate is robust under MRIP. MRIP enables the generation of data in parallel, each replication running an instance of OBM or STS/CSUM.

Despite the expected improvement arising from this combination of approaches (MRIP together with robust method of analysis), batch size determination continues being the great challenge, as it offers an additional burden for determining the batch size to yield acceptable correlation among the batch means. In distributed environment this additional burden can not be parallelized and the user should look for ways of shortening it.

We investigated some alternatives to reduce the batch size determination phase (BSDP). NOBM implements the method of jackknife to estimate correlation coefficient. Estimators based on Jackknife produce less biased estimates than the original ones, but unfortunately it is a time-consuming methodology. Some researchers have the opinion that it is preferable to collect more observations instead of using such kind of unnatural estimators.

We used the rank version of von Neuman's test, since it is less time-consuming, and plugged it in our sequential version of OBM. By using 10 and 20 batches the coverage was very poor, as theoretically exposed in [76], that suggested a minimum of 100 batches. Adding more batches could improve the coverage, but the problem of a long BSDP would not be answered. As we increased the degree of parallelization, we obtained the more accurate coverage. In the method of STS/CSUM, we relaxed the significance level of the test for normality, which permitted a reduction of the BSDP and, once more, increasing the degree of parallelization we have got very accurate results.

According to the truncated Amdahl's law proposed by Pawlikowski and McNickle [102], the maximum achievable speedup depends not only on the sequential part f of every parallel processing, but also on the granularity of consecutive checkpoints. In this particular, OBM is superior to the other CIPs, since granularity can be as fine as the unity. In other words, after the BSDP, each new collected observation can (theoretically) produce a checkpoint (see 5.9.

The methods based on STS, as we showed in chapter 5, must collect, at least, a number of observations equal the number of batches. In our implementation, this number is 10.

Actually, extreme cases of OBM and STS can not be achieved without overloading

the global analyzer. Even so, the maximum achievable speedup is already attractive.

7.3 Conclusions

As focused by many pratictioners of parallel programming, deriving a sequential program into a parallel one is not trivial. MRIP is an interesting and promising approach for increasing the efficiency of simulation experiments of dynamic systems such as communication networks. Akaroa-2, an MRIP implementation, is ease to install, use and extend. Further research in order to implement new sequential procedures that are statistically efficient has received due attention at the University of Canterbury, in Christchurch, New Zealand, at the Technical University of Berlin, Germany, and now at the University of Amazonas, in Manaus, Brazil.

Experimental investigation showed us that for low traffic intensities, NOBM performs satisfactorily, and CSUM should be avoided or never used, since the variability of the confidence interval is very high (see 5.2. For medium values of traffic intensity, or when the run length is long due, for example, to a tight relative precision, NOBM/GW and SBM should be selected preferentially. Concerning the settings for NOBM/GW and SBM that should be made by the user, we claim that those explained in chapter 4 are acceptable for this range of traffic intensity. Sophisticated methods of selecting the spacing of SBM gives no practical improvement.

When simulating queuing systems with high traffic intensity OBM and CSUM should be the best choices among the CIPs studied in this research. If accuracy is an issue, we suggest OBM which requires less observations and is more intuitive.

On the other hand, by simulating a very time-consuming model, CSUM can be selected, since it has been proven to be asymptotically more efficient, and require less observations in presence of more processors. The sequential version of CSUM implemented in this research is an estimator with substantially lower computational cost, if it can be run in an environment of multiple replications in parallel.

7.4 Future research

Further experimental testing of this methodology against data sets with known properties is required. We limited this research to a set of reference queuing models, but it would be advisable to extend the analysis to queuing networks with more complex correlation structure.

Akaroa-2 is still being developed and certainly the global analyzer should be improved to in order to allow finer checkpoint granularity, which could yield higher speedup.

Batch-Means-based procedures deserve further investigation for their simplicity of conception. New ideas as those proposed by Steiger and Wilson [125] can be suitably integrated into a sequential procedure under MRIP. A very challenging investigation should be the integration of a methodology proposed by Pieter Voss [127], which states that is possible to estimate steady-state means from a short, autocorrelated and transient time series.

It would be very welcome a proposal of a more efficient manner of averaging the intermediates estimates coming form different and independent replications. Some authors claim it does really exist, but a superficial investigation gave us poor or, at most identical results.

Both OBM and CSUM can be tuned towards a shorter BSDP, but in this case we should have many processors at our disposal. Moreover, the precise choice of the number of processors that can compensate an imprecise batch selection is an open issue that can be investigated in a future research.

The sequential methods here investigated could be certainly tuned in such a way that they could give speedup close to the one resulted from Amdahl law. So far we have focused on implementing methods with a good coverage.

A

THE SHAPIRO-WILK TEST

i\n	2	3	4	5	6	7	8	9	10	
1	.7071	.7071	.6872	.6646	.6431	.6233	.6052	.5888	.5739	
2	–	.0000	.1677	.2413	.2806	.3031	.3164	.3244	.3291	
3	–	–	–	.0000	.0875	.1401	.1743	.1976	.2141	
4	–	–	–	–	–	.0000	.0561	.0947	.1224	
5	–	–	–	–	–	–	–	.0000	.0399	
i\n	11	12	13	14	15	16	17	18	19	20
1	.5601	.5475	.5359	.5251	.5150	.5056	.4968	.4886	.4808	.4734
2	.3315	.3325	.3325	.3318	.3306	.3290	.3273	.3253	.3232	.3211
...	...	...	...	...	...	...	...	...	...	...

Tab. 21: Coefficients a_{in} for the Shapiro-Wilk test

	level								
n	**0.01**	**0.02**	**0.05**	**0.10**	**0.50**	**0.90**	**0.95**	**0.98**	**0.99**
3	0.753	0.756	0.767	0.789	0.959	0.998	0.999	1.000	1.000
4	0.687	0.707	0.748	0.792	0.935	0.987	0.992	0.996	0.997
5	0.686	0.715	0.762	0.806	0.927	0.979	0.986	0.991	0.993
6	0.713	0.743	0.788	0.826	0.927	0.974	0.981	0.986	0.989
7	0.730	0.760	0.803	0.838	0.928	0.972	0.979	0.985	0.988
8	0.749	0.778	0.818	0.851	0.932	0.972	0.978	0.984	0.987
9	0.764	0.791	0.829	0.859	0.935	0.972	0.978	0.984	0.986
10	**0.781**	0.806	0.842	0.869	0.938	0.972	**0.978**	0.983	0.986
...	...	...	...	...	...	...	...	...	...

Tab. 22: Critical points for the Shapiro-Wilk test

Small values of Shapiro-Wilk statistic W are significant. By testing normality among the random variables A_i, the asymptotic scaled sum of $T_i(t)$ for each batch by means of in the method of *Standardized Time Series* (see Sec. 2.3.5), we chose the 0.95 level of significance. Since we chose a number of batches equal 10, the hypothesis of normality will be rejected if $W < 0.978$.

When we wanted to force imprecision of selection of batch size, we chose a lower level of significance (e.g. 0.01) for the test of normality. This way, test for normality succeeds when the Shapiro-Wilk statistics $W > 0.781$.

B

THE RANK VON NEUMANN TEST

$n\backslash\alpha$	.005	.010	.025	.050	.100
10	.62	.72	.89	1.04	1.23
11	.67	.77	.93	1.08	1.26
12	.71	.81	.96	1.11	1.29
13	.74	.84	1.00	1.14	1.32
14	.78	.87	1.03	1.17	1.34
15	.81	.90	1.05	1.19	1.36
16	.84	.93	1.08	1.21	1.38
17	.87	.96	1.10	1.24	1.40
18	.89	.98	1.13	1.26	1.41
19	.92	1.01	1.15	1.27	1.43
20	.94	1.03	1.17	1.29	1.44
21	.96	1.05	1.18	1.31	1.45
22	.98	1.07	1.20	1.32	1.46
23	1.00	1.09	1.22	1.33	1.48
24	1.02	1.10	1.23	1.35	1.49
25	1.04	1.12	1.25	1.36	1.50
26	1.05	1.13	1.26	1.37	1.51
27	1.07	1.15	1.27	1.38	1.51
28	1.08	1.16	1.28	1.39	1.52
29	1.10	1.18	1.30	1.40	1.53
30	1.11	1.19	1.31	1.41	1.54
32	1.13	1.21	1.33	1.43	1.55
34	1.16	1.23	1.35	1.45	1.57
36	1.18	1.25	1.36	1.46	1.58
38	1.20	1.27	1.38	1.48	1.59
40	1.22	1.29	1.39	1.49	1.60
42	1.22	1.29	1.41	1.50	1.61
44	1.25	1.32	1.42	1.51	1.62
46	1.27	1.33	1.43	1.52	1.63
48	1.28	1.35	1.45	1.53	1.63
50	1.29	1.36	1.46	1.54	1.64
55	1.33	1.39	1.48	1.56	1.66
60	1.35	1.41	1.50	1.58	1.67
65	1.38	1.43	1.52	1.60	1.68
70	1.40	1.45	1.54	1.61	1.70
75	1.42	1.47	1.55	1.62	1.71
80	1.44	1.49	1.57	1.64	1.71
85	1.45	1.50	1.58	1.65	1.72
90	1.47	1.52	1.59	1.66	1.73
95	1.48	1.53	1.60	1.66	1.74
100	1.49	1.55	1.61	1.67	1.74
...	...	...	...	...	...

Tab. 23: Critical points for the Rank von Neumann test

C

QUEUING MODELS

We ran experiments on the waiting time process in an $M/M/1$ queue. In many practical situations, the exponential assumptions may be rather limiting, especially concerning service times being distributed exponentially.

One possibility in communication traffic analysis is the modeling of systems with batch arrival and departure processes [68]. Therefore, we also ran experiments on the waiting time process in an $M/E_k/1$ queue, which provides a model for batch arrivals [77]. The symbol E_k represents Erlangian services times with k stages. Erlang-r distribution approximates nonexponential distributions with coefficient of variation (C_x) less than one (mechanism of service in an $M/M/1$ queue has unity coefficient of variation).

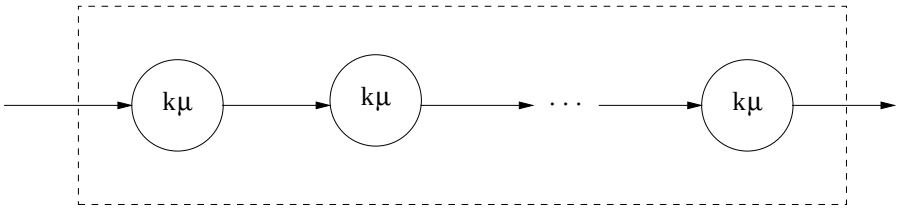


Fig. 35: Erlangian system with k stages

A customer can just begin being served by the first stage when the customer in service left the last stage. We chose to work with 4 stages, so then the coefficient of variation equals $1/\sqrt{4}$. There was no special reason for this choice, except to achieving a coefficient of variation between 0 and 1.

To get a queuing system with coefficient of variation greater than unity, we resorted to hyperexponential distribution with 2 stages. A customer goes to stage j with probability q_j.

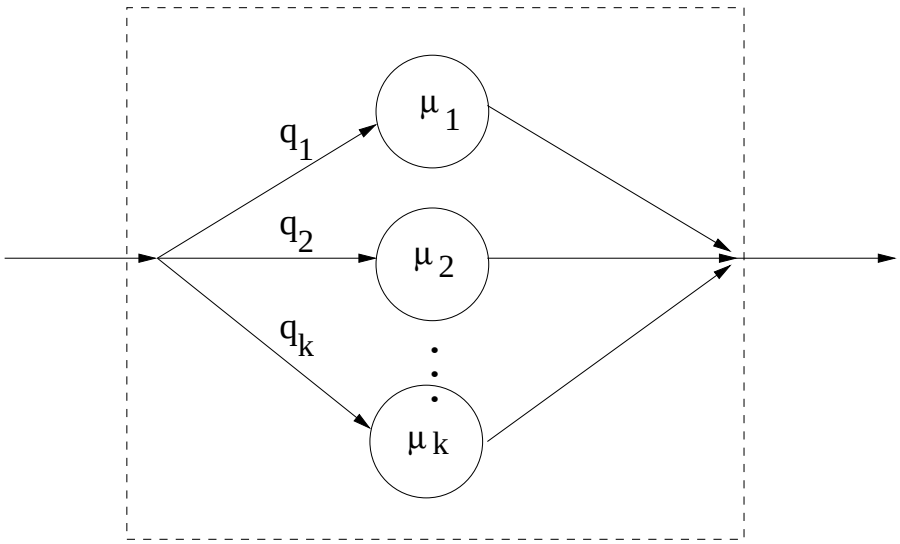


Fig. 36: Hyperexponential system with k stages

The parameter q_i and μ were adjusted to yield a coefficient of variation equal to $\sqrt{2}$ The fourth queuing system we used was an $M/D/1$, where the service time is a fixed, deterministic quantity. This is actually a reasonable model for the processing of fixed length packets in a communication network [110].

Let λ be the arrival rate of each of these queues, μ be the service rate, and ρ be the traffic intensity. Let X_n be the waiting time of the nth customer. If $\rho < 1$, there exists a steady-state random variable X such that $X_n \Rightarrow X$ as $n \to \infty$. We are particularly interested in the performance measure of mean waiting time given by :

Model	C_s	$E[X]$
$M/D/1$	0	$\frac{\rho\mu}{2(1-\rho)}$
$M/E_4/1$	$1/\sqrt{4}$	$\frac{\rho(1+1/k)}{2\mu(1-\rho)}$
$M/M/1$	1	$\frac{\rho}{\mu(1-\rho)}$
$M/H_2/1$	$\sqrt{2}$	$\frac{\rho(1+{C_x}^2)}{2\mu(1-\rho)}$

D

NOBM PERFORMANCE

The following pages present the results of sequential coverage analysis by applying NOBM, when simulating the average waiting time of the following queing systems : $M/D/1$, $M/E_4/1$, $M/M/1$ and $M/H_2/1$ (see App. C for more details of their settings). We varied traffic intensity ($\rho = 0.1, 0.5,$ and 0.9), confidence level ($\eta = 0.1$ to 0.95) and degree of parallelization (P = 1, 2 and 6).

We constructed a coverage function for each combination of these factors and, additionally, we assed the average run length (in units of 1000 observations) and CoV{H}, a measure of stability of confidence interval half-width. Graphics of the same type of MOE were plotted using the same unit, in such a way one can compare them visually. A few graphics corresponding to average run length of very low confidence level and traffic intensity, do not appear, since they used much less than 1000 observations.

The end of transient period was found by means of a stationarity test proposed by Schruben et al. [121], already implemented in Akaroa-2. Observations of transient period are discarded, and the stochastic process being simulated is considered to be a sample from a covariance stationary process.

Initially, batch size was selected to be equal 100 observations. Independence among batch means was pursued by using jackknife estimators of correlation coefficients. There was, initially, 100 batches, but after finding the optimal batch size that yielded almost uncorrelated batch means, observations were reorganized into 25 batches, according to [114], which resulted in a batch size four times larger.

At consecutive checkpoints, sample mean and sample variance were calculated and sent to a global analyzer, together with the sample size. The number of batches is kept fixed. The simulation experiment stopped when the relative precision (5%) was achieved two times consecutively.

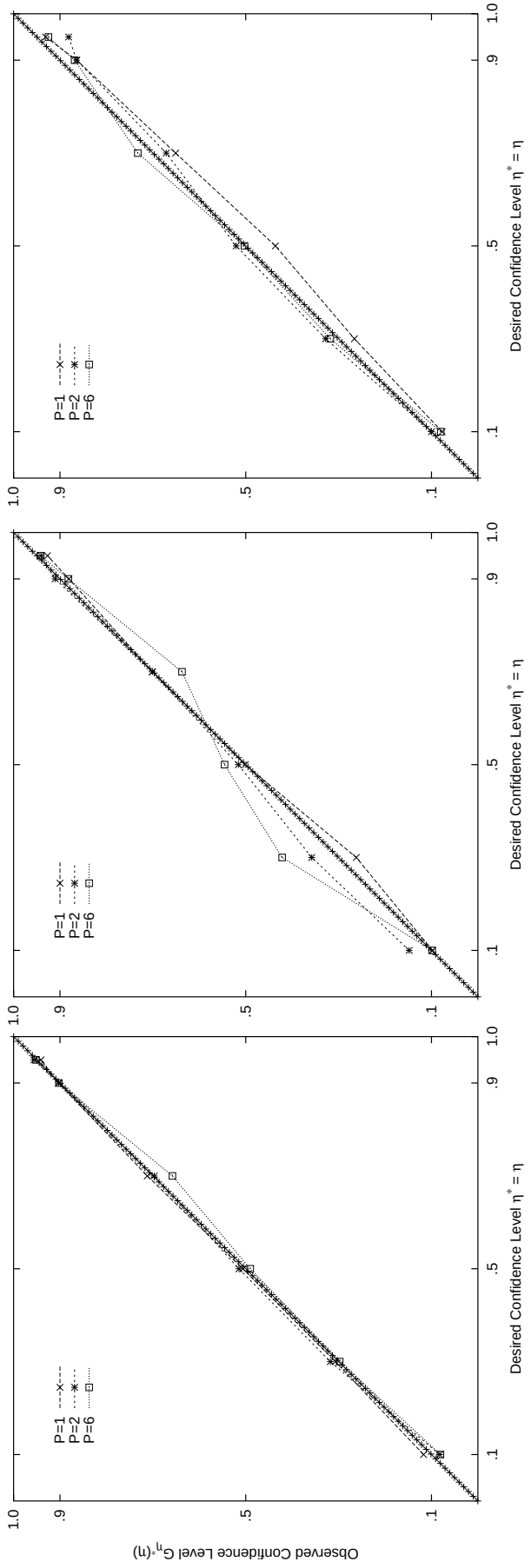


Fig. 37: NOBM : coverage function for $M/D/1$ and $\rho = 0.1, 0.5, 0.9$

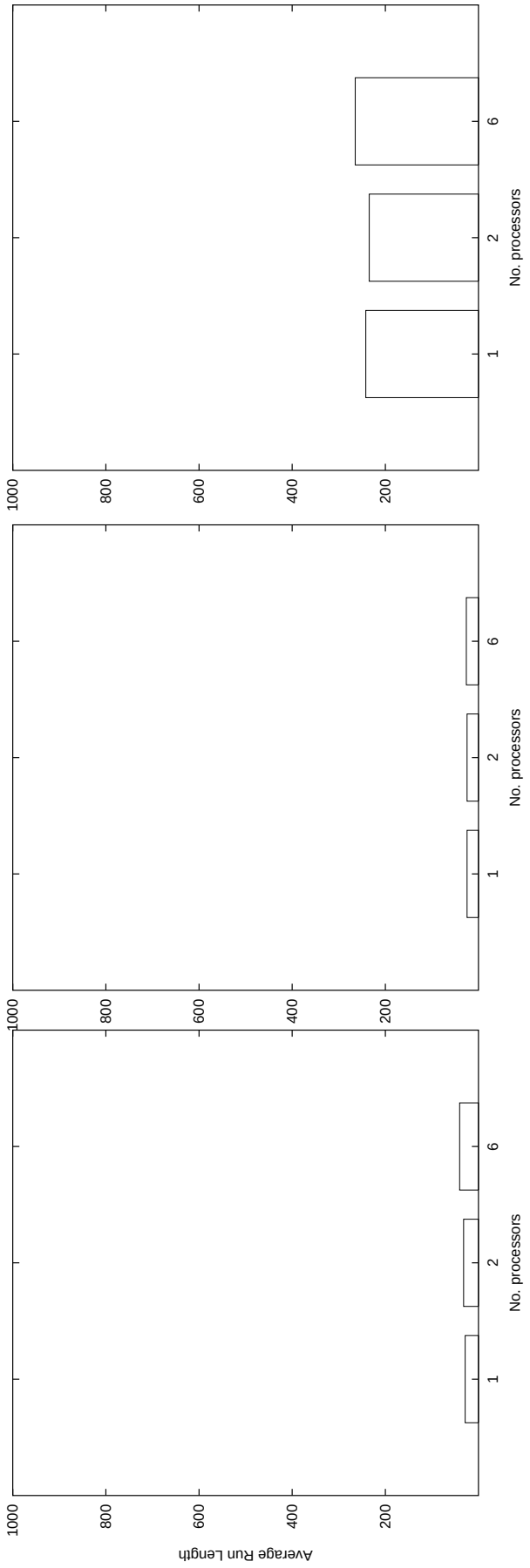


Fig. 38: NOBM : average run length for $M/D/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

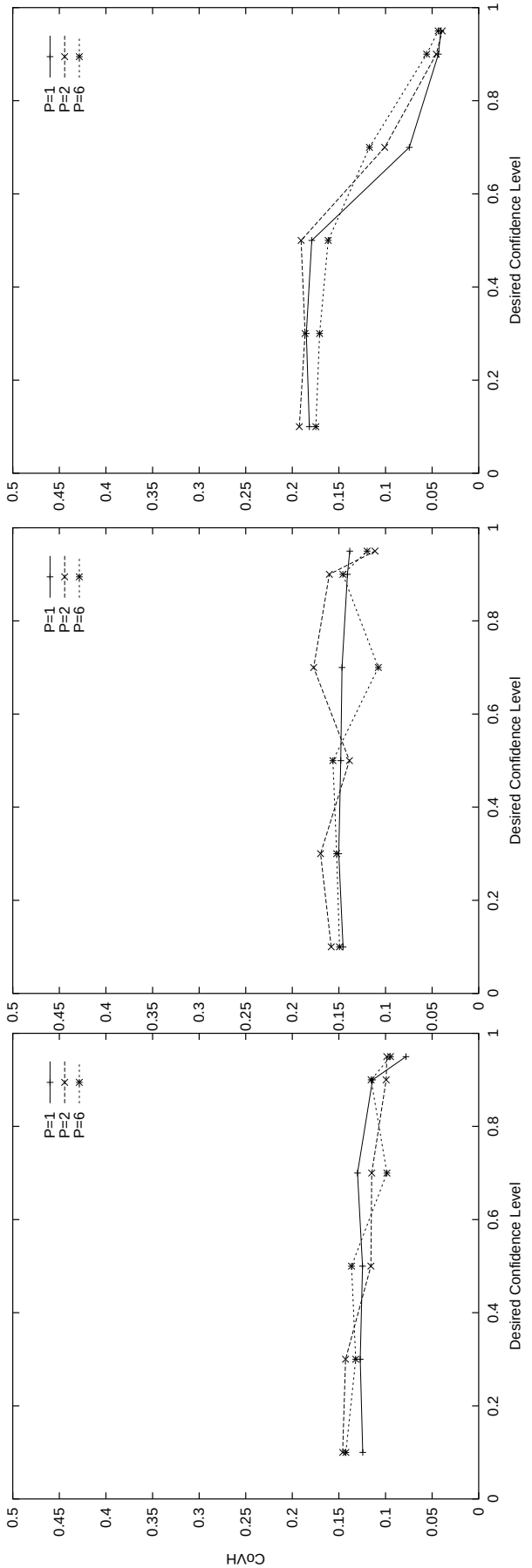


Fig. 39: NOBM : CoV{H} for $M/D/1$ and $\rho = 0.1, 0.5, 0.9$

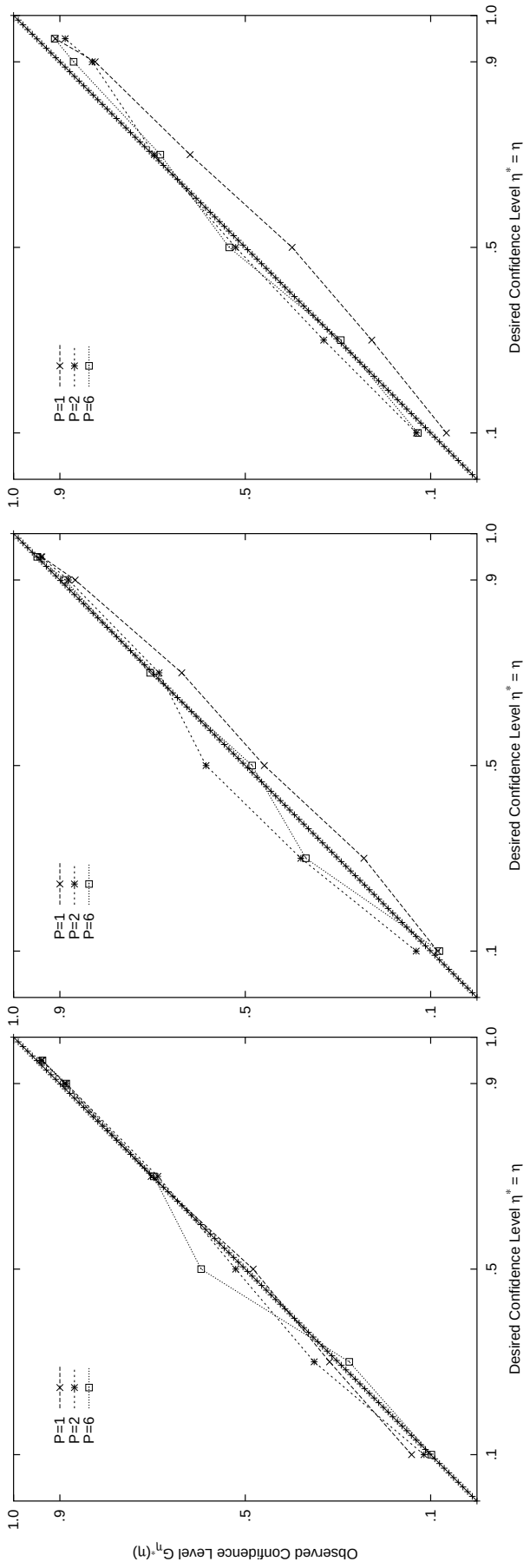


Fig. 40: NOBM : coverage function for $M/E_4/1$

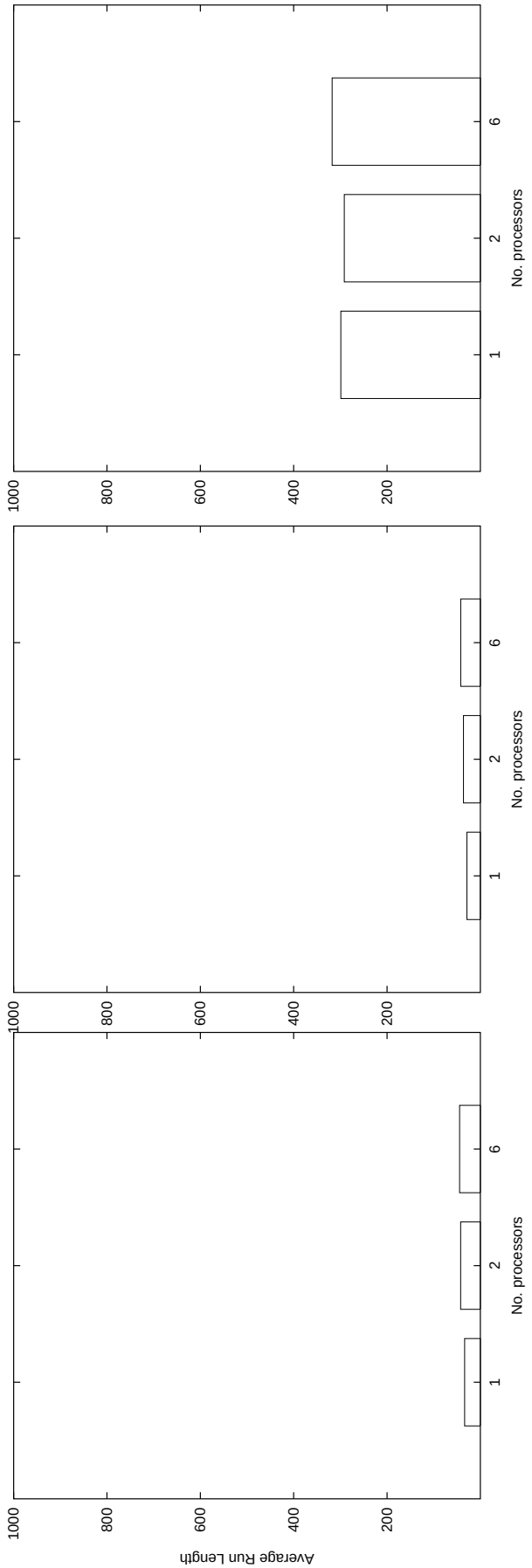


Fig. 41: NOBM : average run length for $M/E_4/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

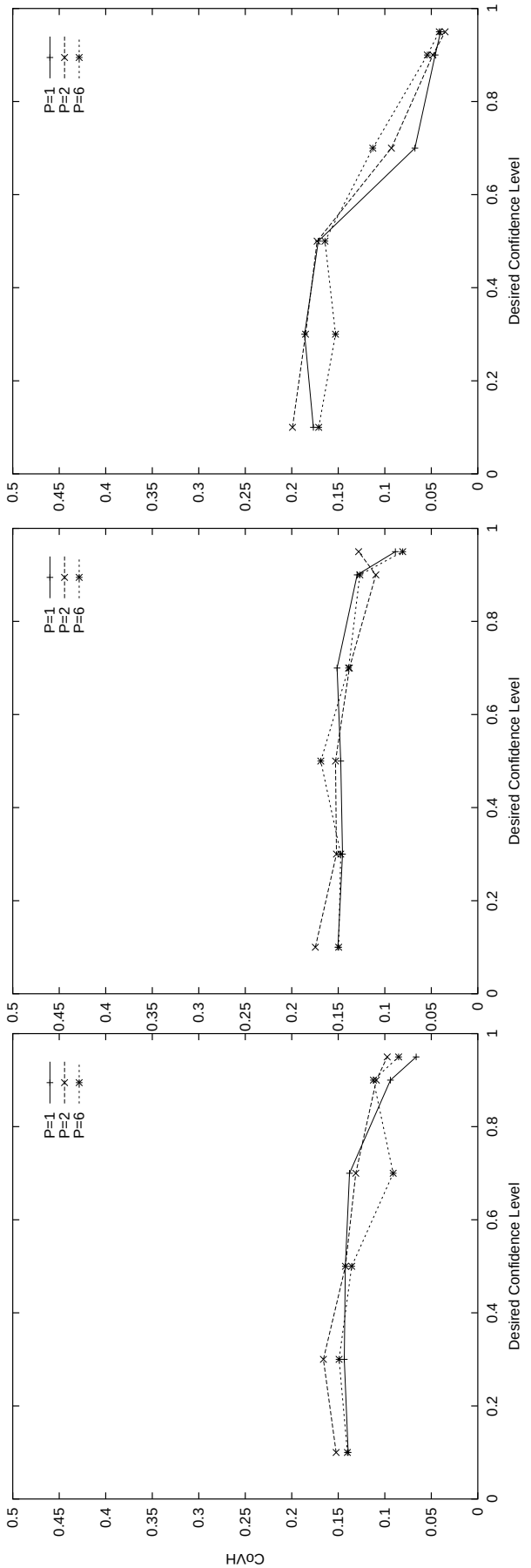


Fig. 42: NOBM : CoV{H} for $M/E_4/1$

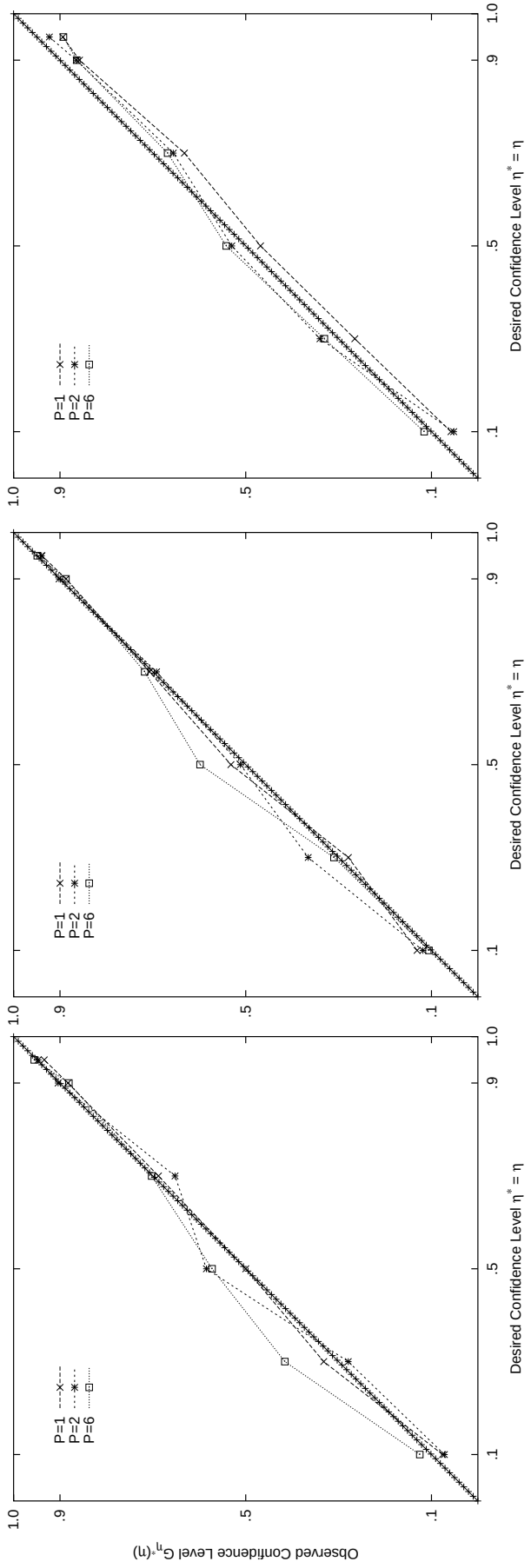


Fig. 43: NOBM : coverage function for $M/M/1$

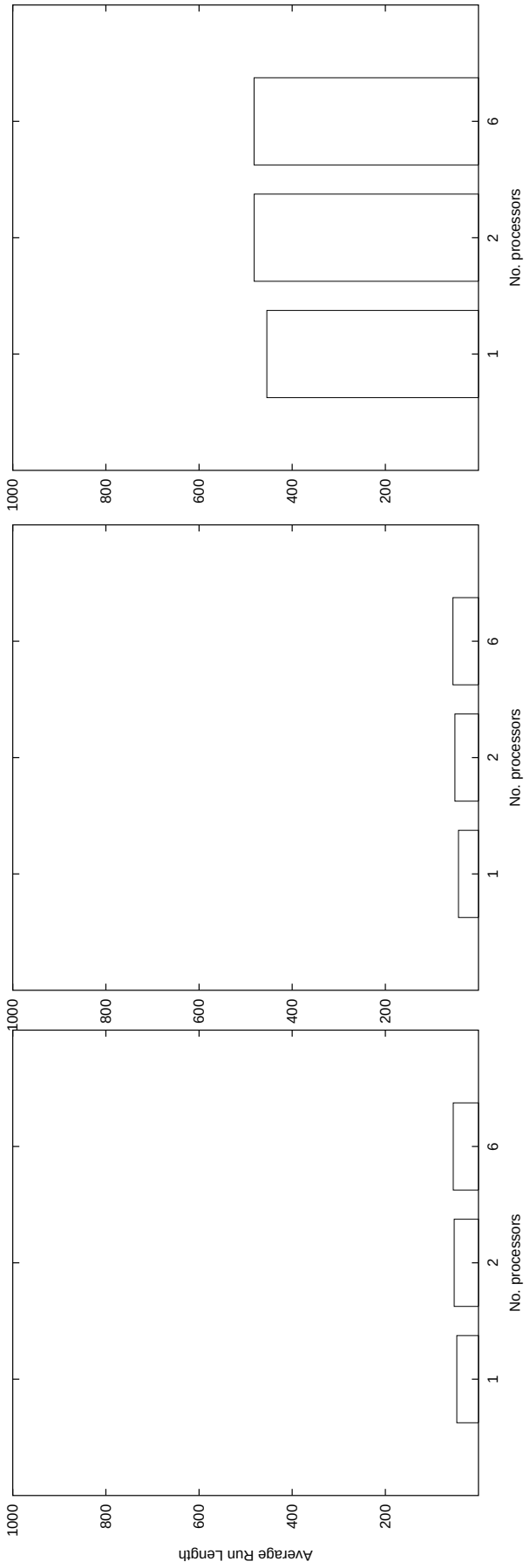


Fig. 44: NOBM : average run length for $M/M/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

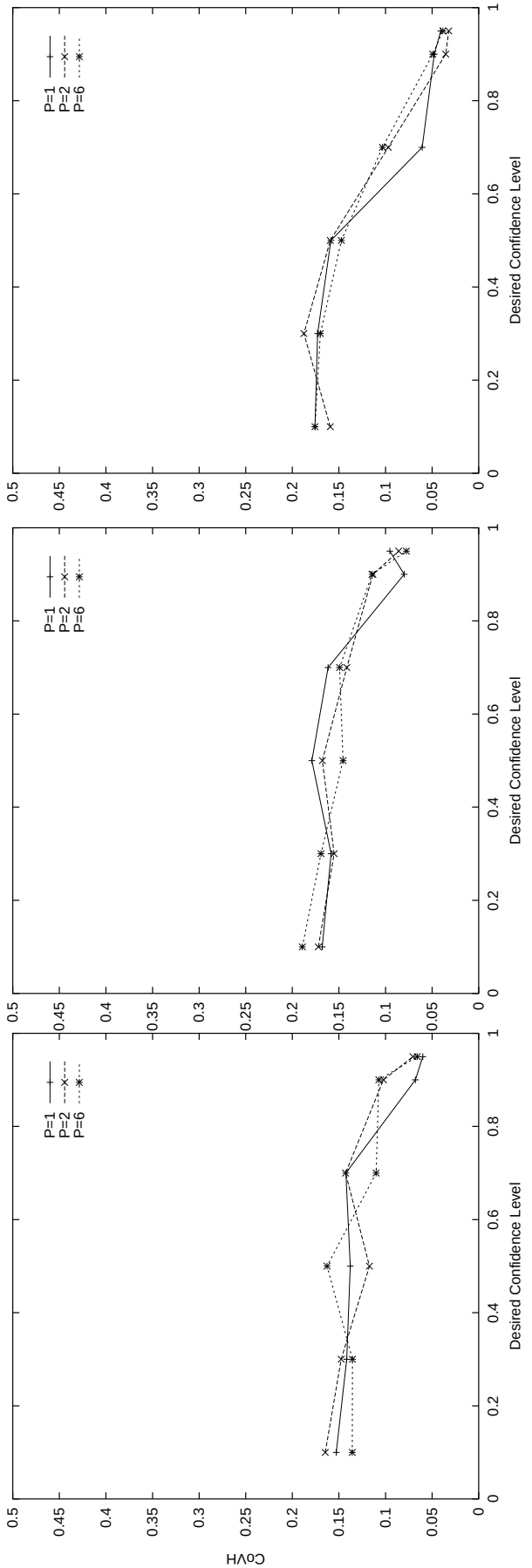


Fig. 45: NOBM : CoV{H} for $M/M/1$

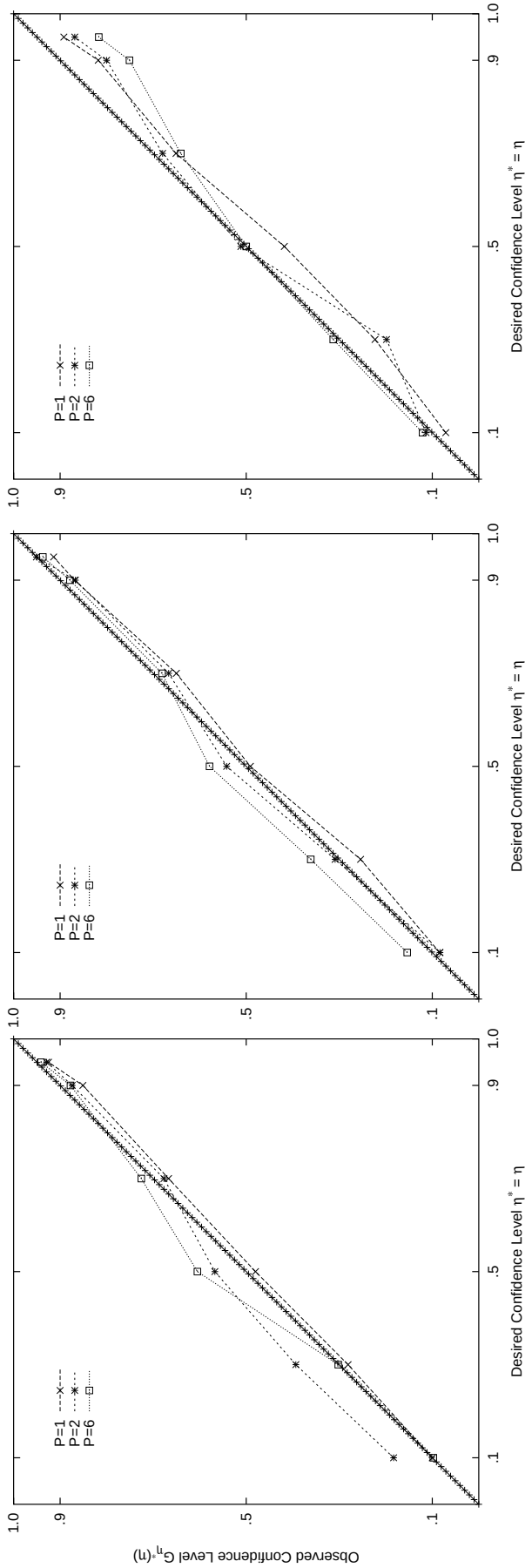


Fig. 46: NOBM : coverage function for $M/H_2/1$

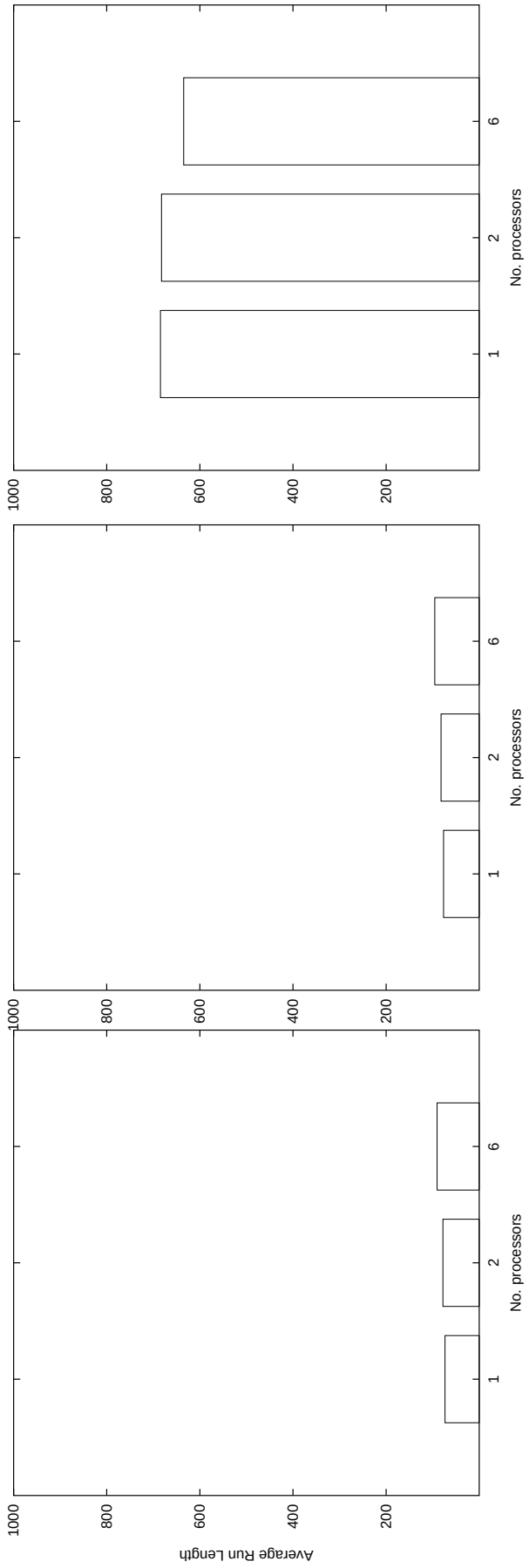


Fig. 47: NOBM : average run length for $M/H_2/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

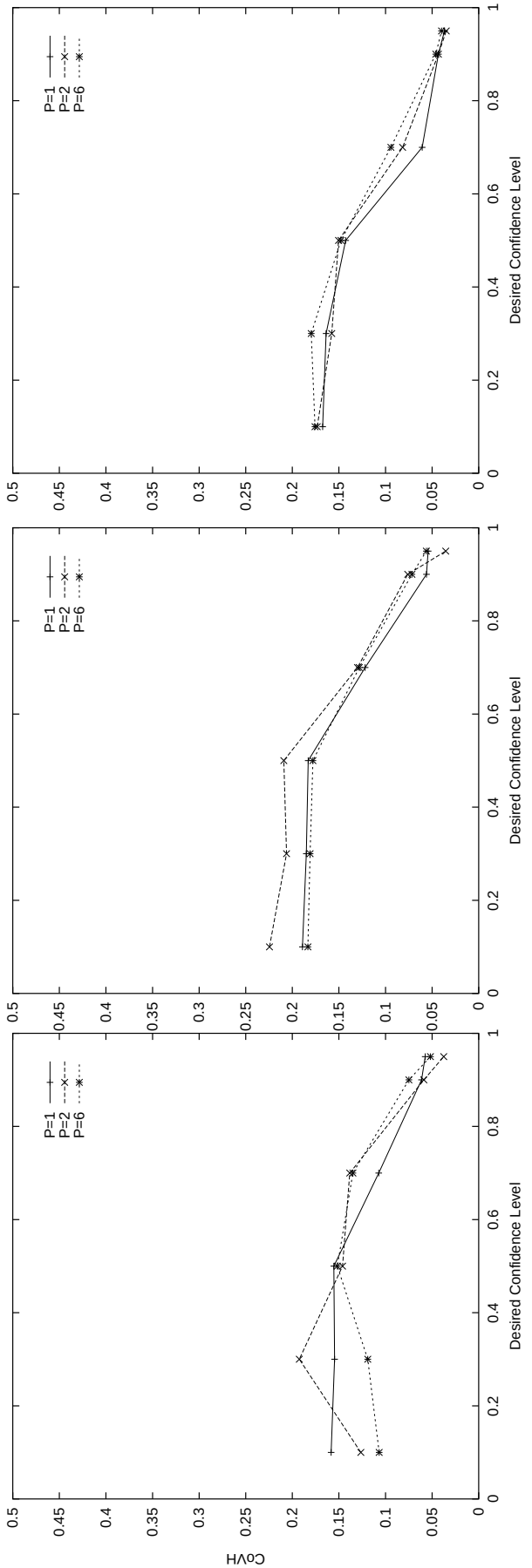


Fig. 48: NOBM : CoV{H} for $M/H_2/1$

E

NOBM/GW PERFORMANCE

The following pages present the results of sequential coverage analysis, by applying NOBM/GW, when simulating the average waiting time of the following queing systems : $M/D/1$, $M/E_4/1$, $M/M/1$ and $M/H_2/1$ (see App. C for more details of their settings). We varied traffic intensity ($\rho = 0.1, 0.5$, and 0.9), confidence level ($\eta = 0.1$ to 0.95) and degree of parallelization (P = 1, 2 and 6).

We constructed a coverage function for each combination of these factors and, additionally, we assed the average run length (in units of 1000 observations) and CoV{H}, a measure of stability of confidence interval half-width. Graphics of the same type of MOE were plotted using the same unit, in such a way one can compare them visually. A few graphics corresponding to average run length of very low confidence level and traffic intensity, do not appear, since they used much less than 1000 observations.

The end of transient period was found by means of a stationarity test proposed by Schruben et al. [121], already implemented in Akaroa-2. Observations of transient period are discarded, and the stochastic process being simulated is considered to be a sample from a covariance stationary process.

Initially, batch size was selected to be equal 100 observations. Independence among batch means was pursued by using jackknife estimators of correlation coefficients. There was, initially, 100 batches, but after finding the optimal batch size that yielded almost uncorrelated batch means, observations were reorganized into 25 batches, according to [114], which resulted in a batch size four times larger.

At consecutive checkpoints, sample mean and sample variance were calculated and sent to a global analyzer, together with the sample size. The number of batches is increased by 2 whenever the relative precision is not achieved. The simulation experiment stopped when the relative precision (5%) was achieved two times consecutively.

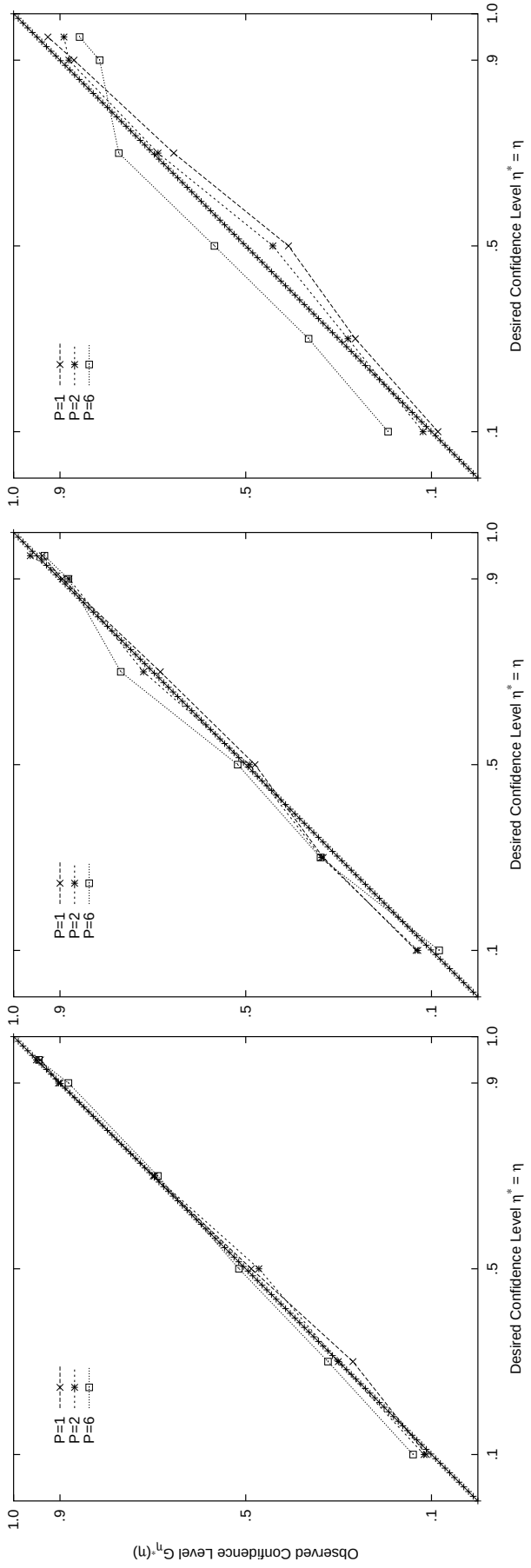


Fig. 49: NOBM/GW : coverage function for $M/D/1$ and $\rho = 0.1, 0.5, 0.9$

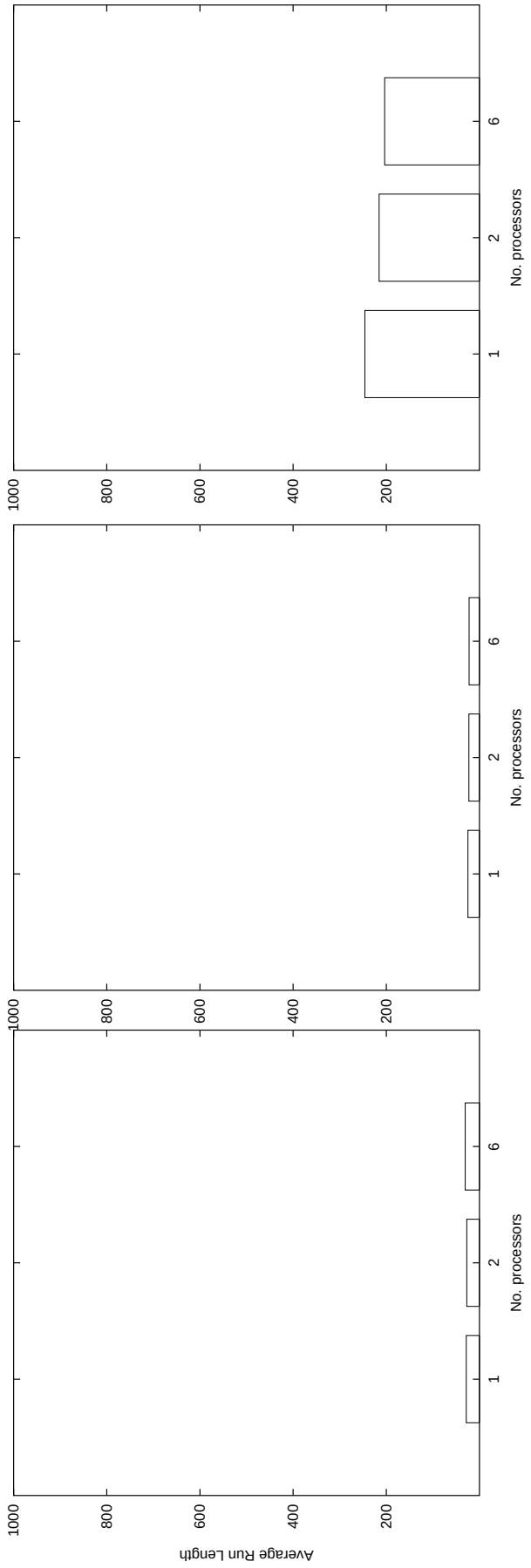


Fig. 50: NOBM/GW : average run length for $M/D/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

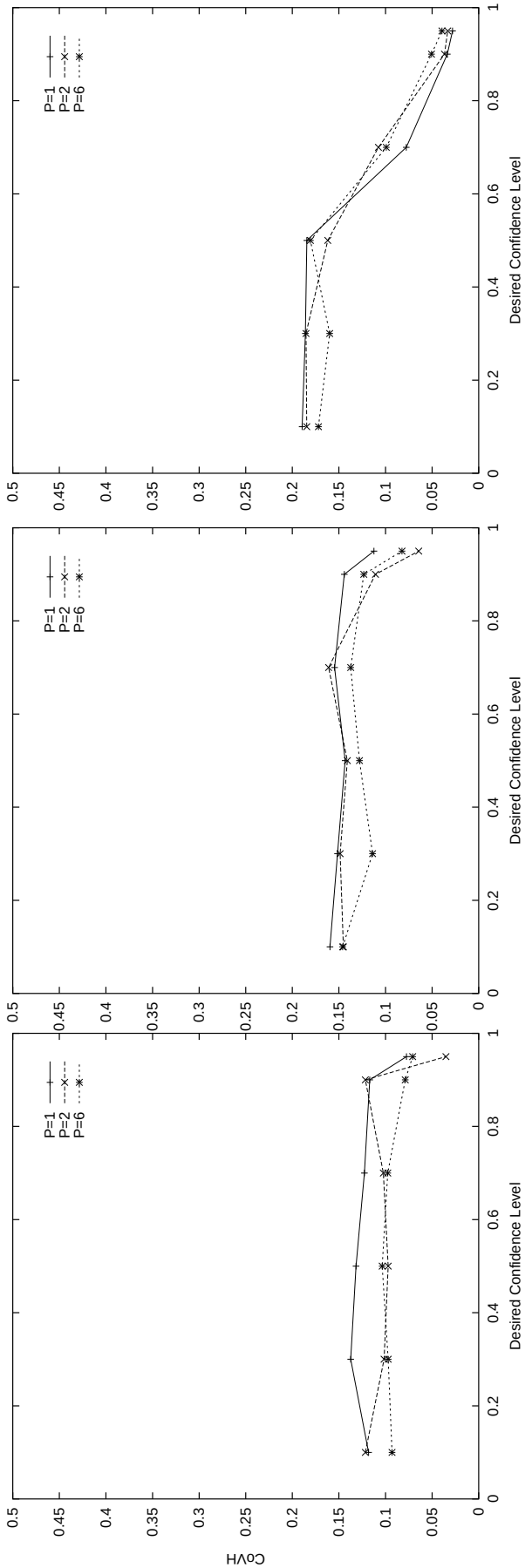


Fig. 51: NOBM/GW : CoV{H} for $M/D/1$ and $\rho = 0.1, 0.5, 0.9$

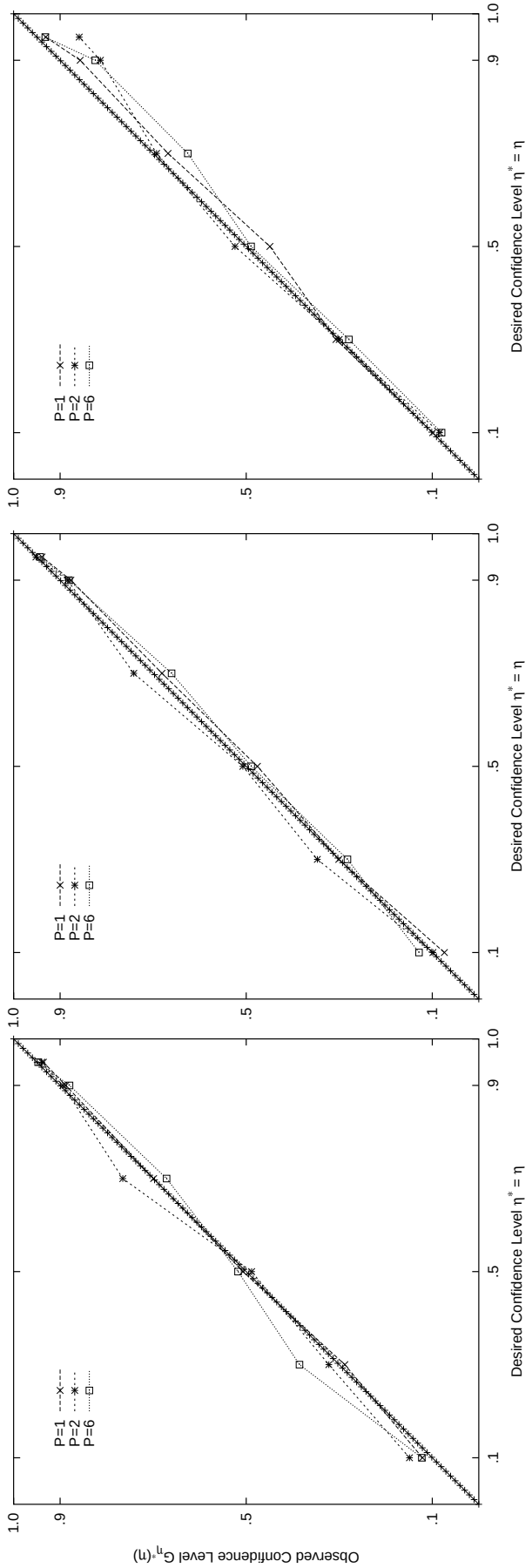


Fig. 52: NOBM/GW : coverage function for $M/E_4/1$ and $\rho = 0.1, 0.5, 0.9$

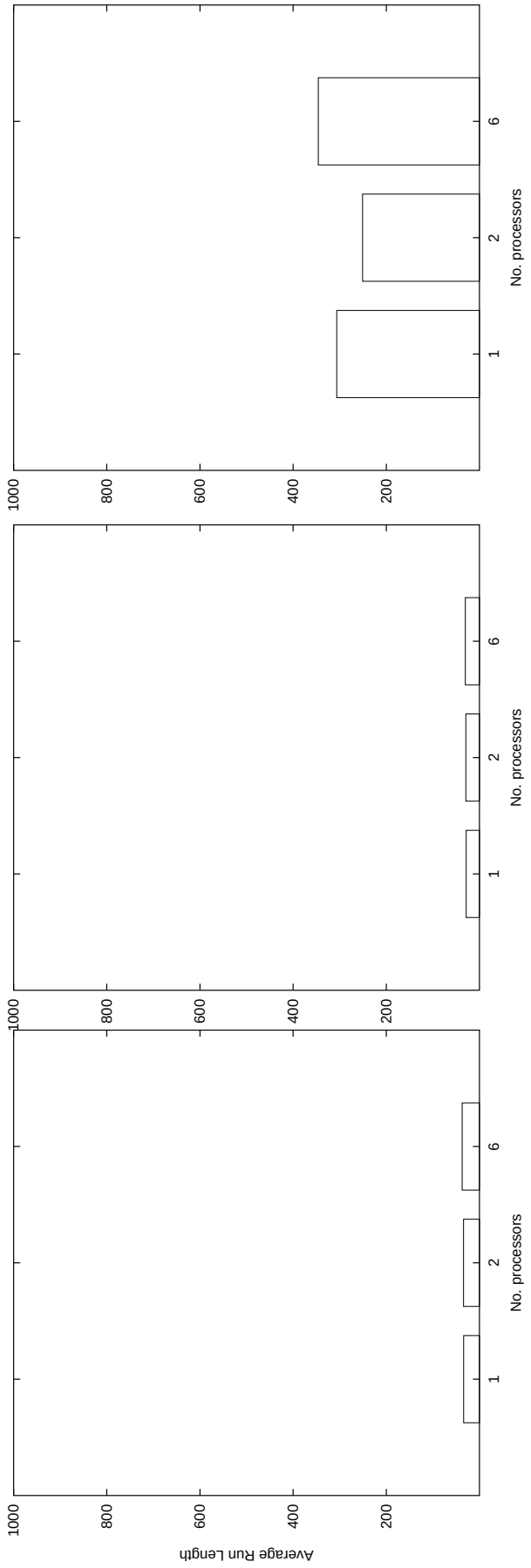


Fig. 53: NOBM/GW : average run length for $M/E_4/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

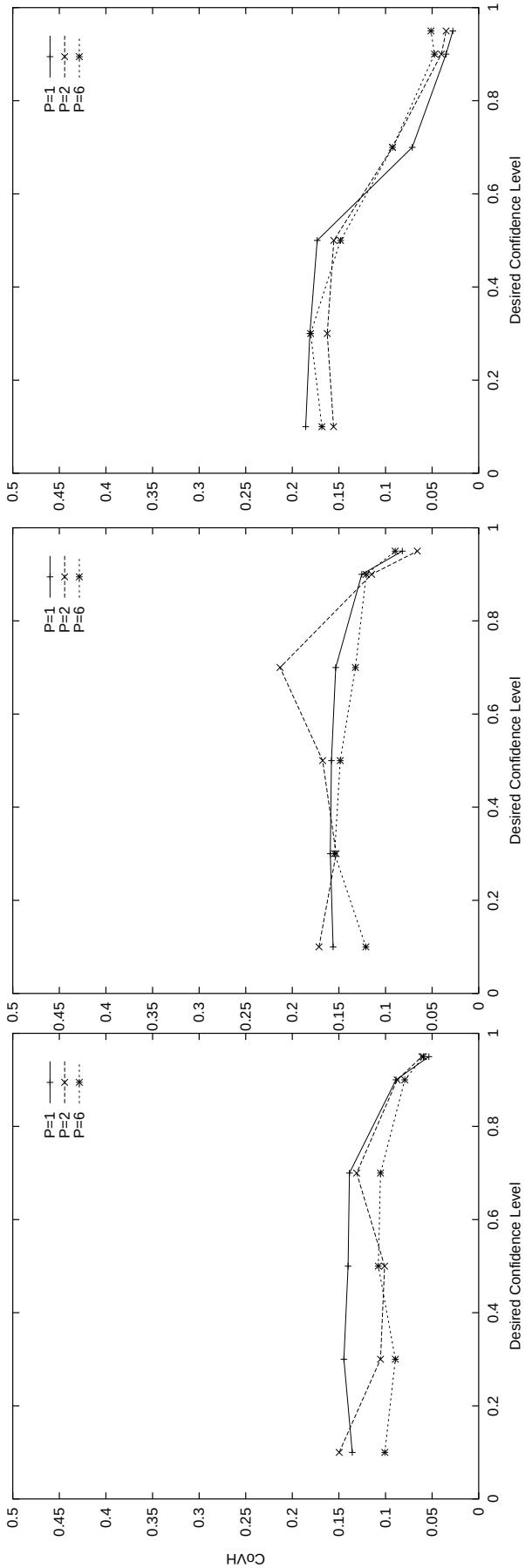


Fig. 54: NOBM/GW : CoV{H} for $M/E_4/1$ and $\rho = 0.1, 0.5, 0.9$

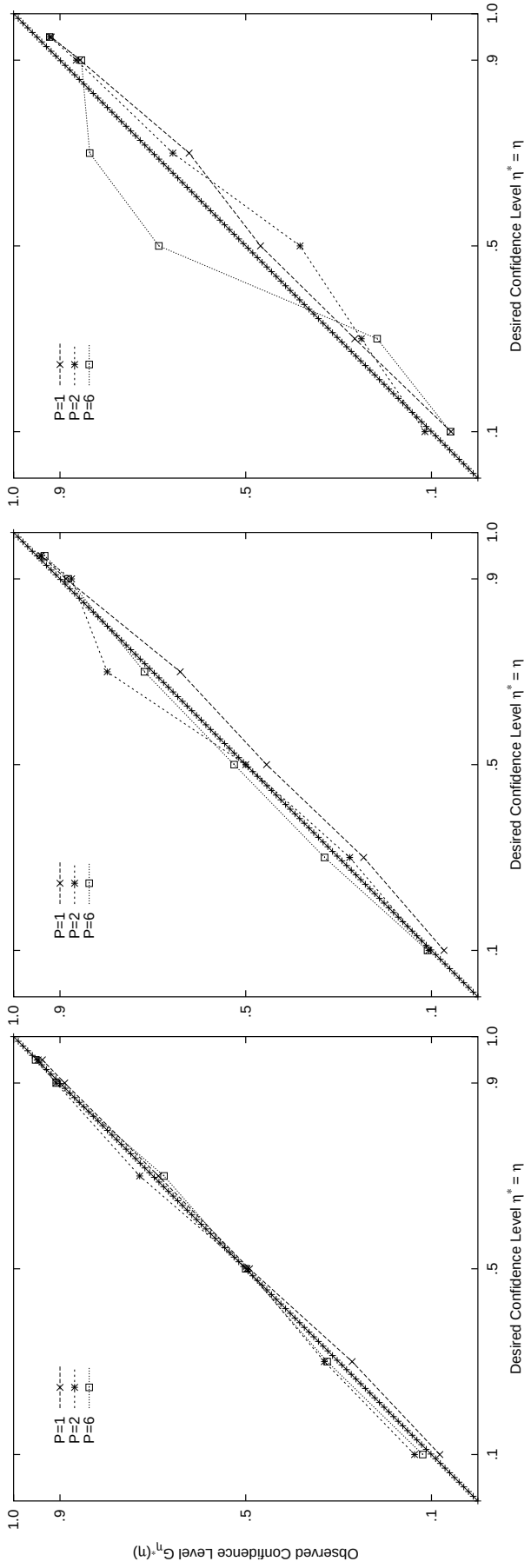


Fig. 55: NOBM/GW : coverage function for $M/M/1$ and $\rho = 0.1, 0.5, 0.9$

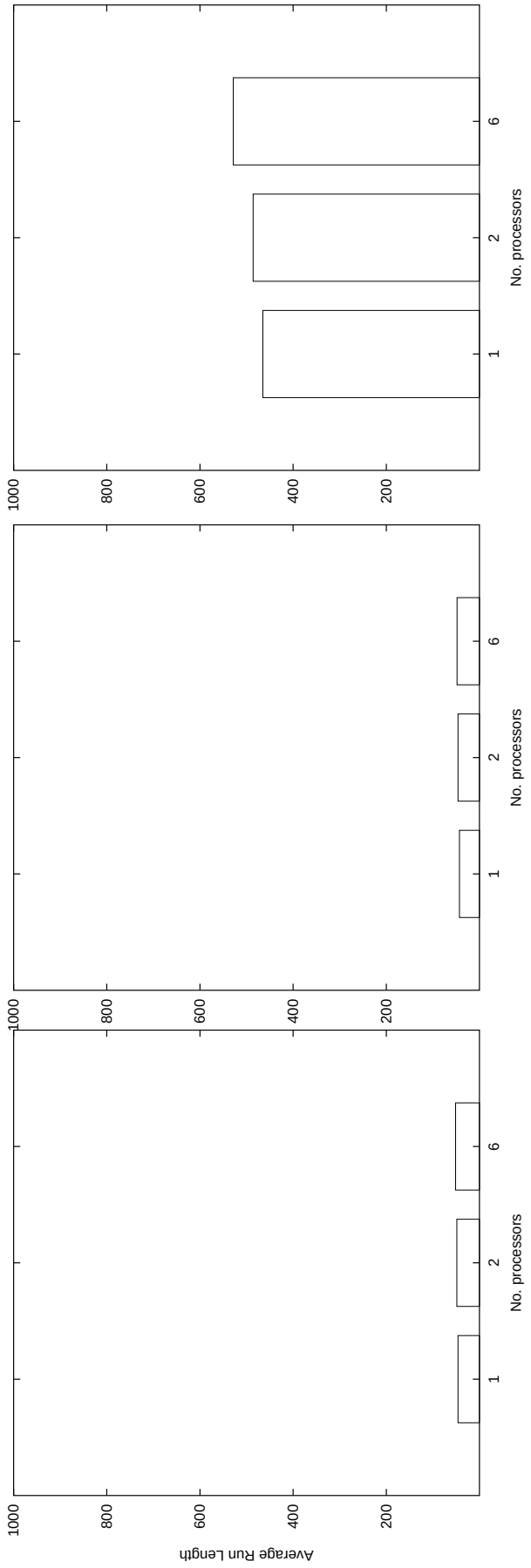


Fig. 56: NOBM/GW : average run length for $M/M/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

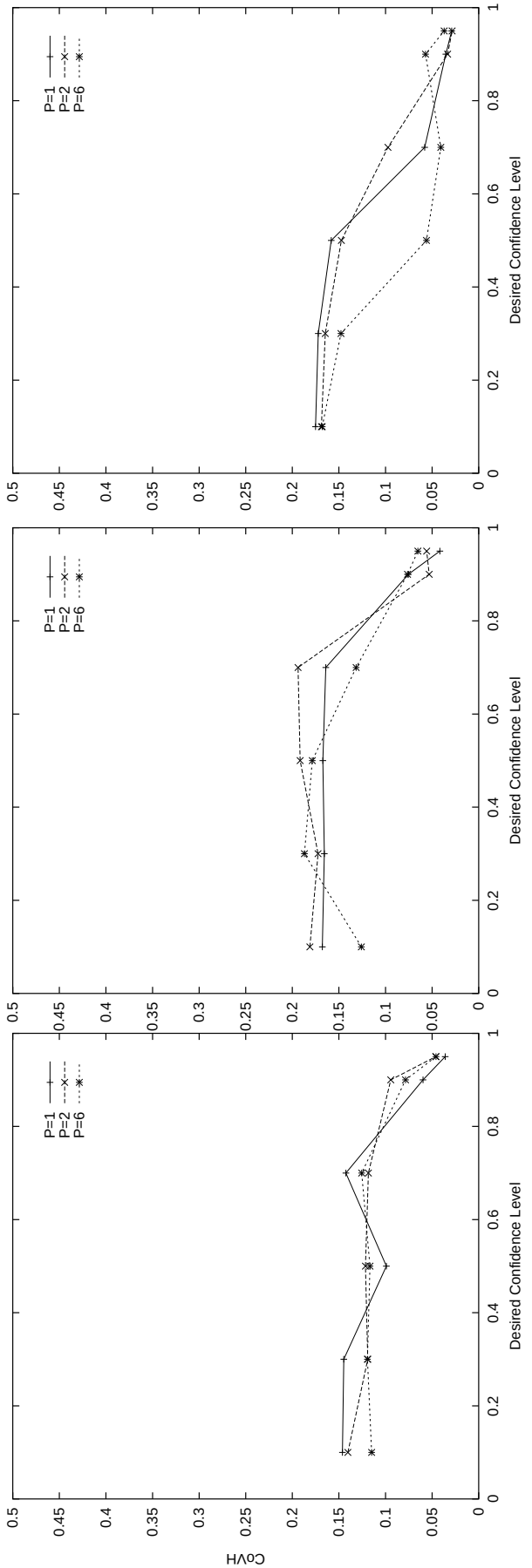


Fig. 57: NOBM/GW : CoV{H} for $M/M/1$ and $\rho = 0.1, 0.5, 0.9$

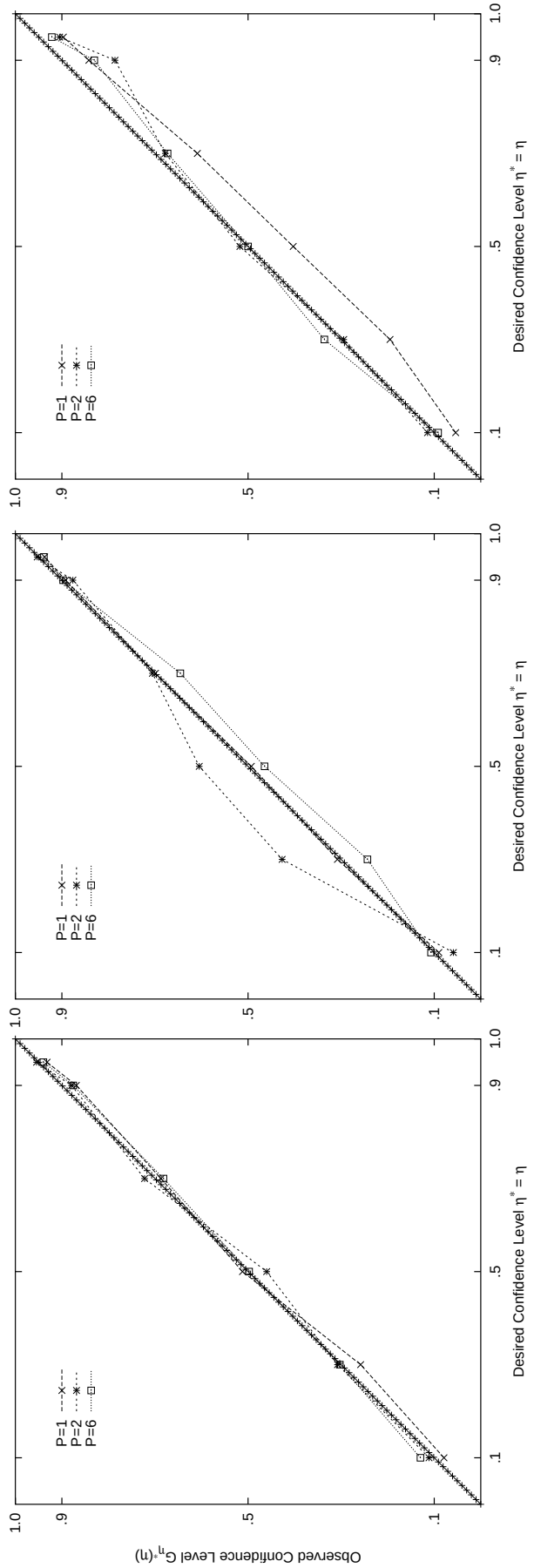


Fig. 58: NOBM/GW : coverage function for $M/H_2/1$ and $\rho = 0.1, 0.5, 0.9$

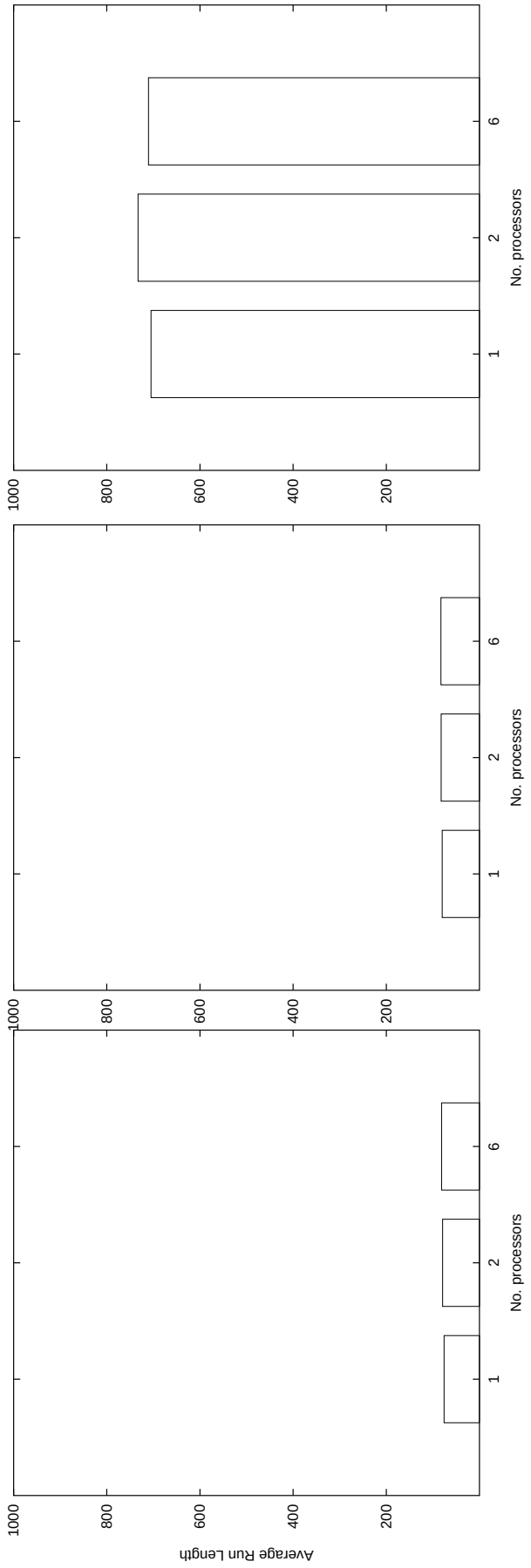


Fig. 59: NOBM/GW : average run length for $M/H_2/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

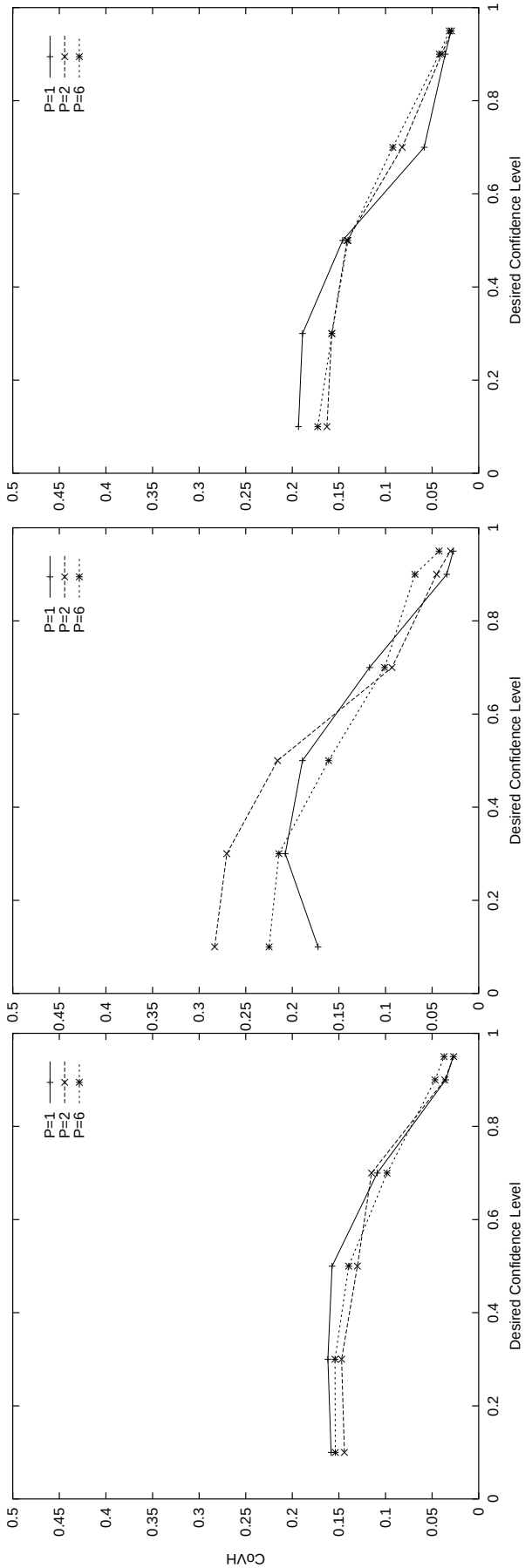


Fig. 60: NOBM/GW : CoV{H} for $M/H_2/1$ and $\rho = 0.1, 0.5, 0.9$

F

SBM PERFORMANCE

The following pages present the results of sequential coverage analysis, by applying SBM, when simulating the average waiting time of the following queing systems : $M/D/1$, $M/E_4/1$, $M/M/1$ and $M/H_2/1$ (see App. C for more details of their settings). We varied traffic intensity ($\rho = 0.1, 0.5,$ and 0.9), confidence level ($\eta = 0.1$ to 0.95) and degree of parallelization (P = 1, 2 and 6).

We constructed a coverage function for each combination of these factors and, additionally, we assed the average run length (in units of 1000 observations) and CoV{H}, a measure of stability of confidence interval half-width. Graphics of the same type of MOE were plotted using the same unit, in such a way one can compare them visually. A few graphics corresponding to average run length of very low confidence level and traffic intensity, do not appear, since they used much less than 1000 observations.

The end of transient period was found by means of a stationarity test proposed by Schruben et al. [121], already implemented in Akaroa-2. Observations of transient period are discarded, and the stochastic process being simulated is considered to be a sample from a covariance stationary process.

Given an initial bath size m=100, the last s of every m observations are discarded. In these experiment, we selected s=0.2m. Independence among batch means was pursued by using jackknife estimators of correlation coefficients. whenever test for correlation fails, multiple of (m - s) observations were grouped, and more observations collected. There was, initially, 100 batches, but after finding the optimal batch size that yielded almost uncorrelated batch means, observations were reorganized into 25 batches, according to [114].

At consecutive checkpoints, sample mean and sample variance were calculated and sent to a global analyzer, together with the sample size. The number of batches is increased by 2 whenever the relative precision is not achieved. The simulation experiment stopped when the relative precision (5%) was achieved two times consecutively.

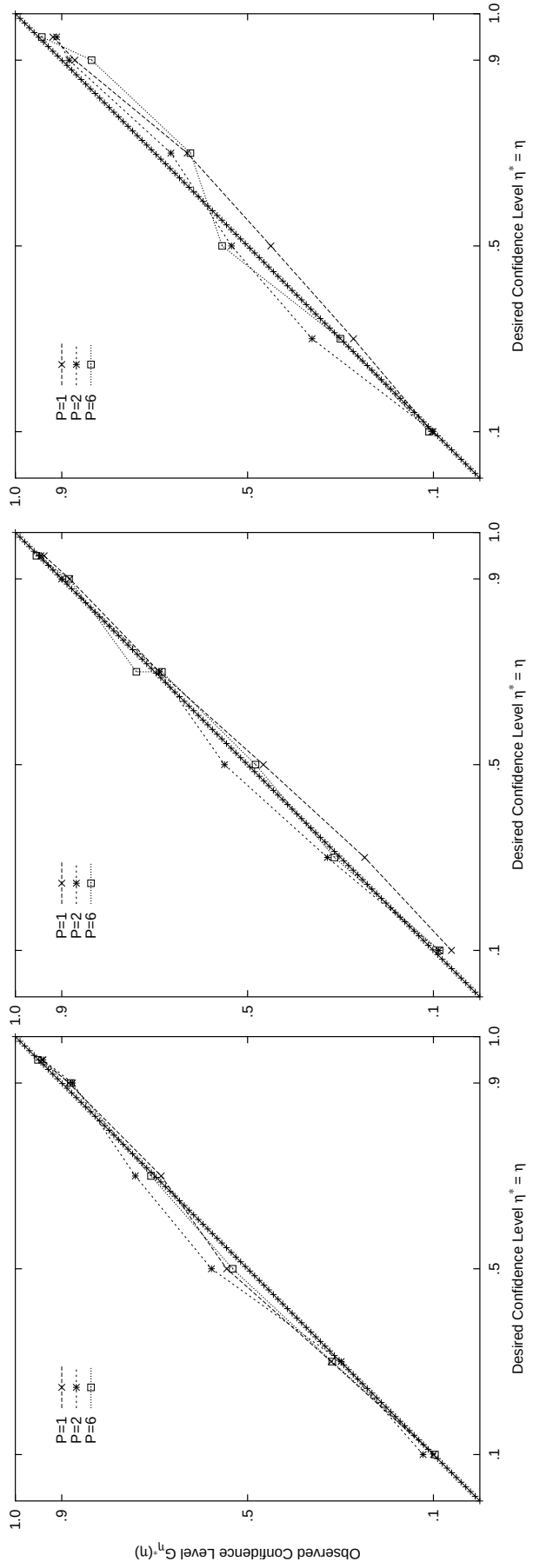


Fig. 61: SBM : coverage function for $M/D/1$ and $\rho = 0.1, 0.5, 0.9$

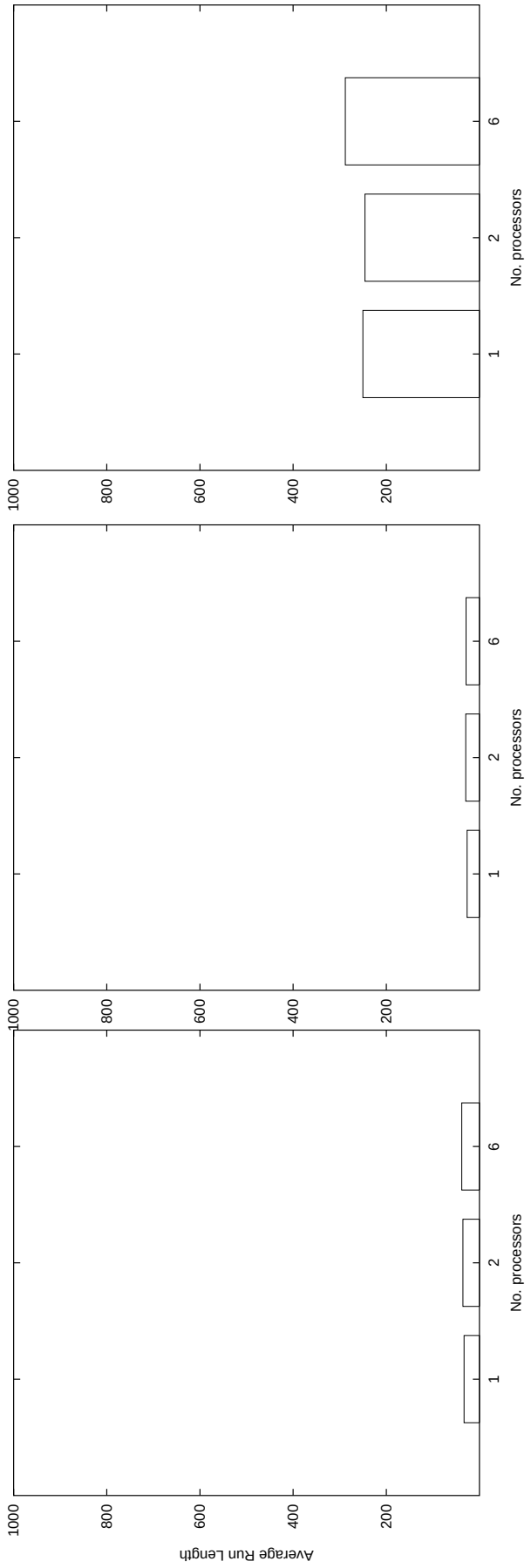


Fig. 62: SBM : average run length for $M/D/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

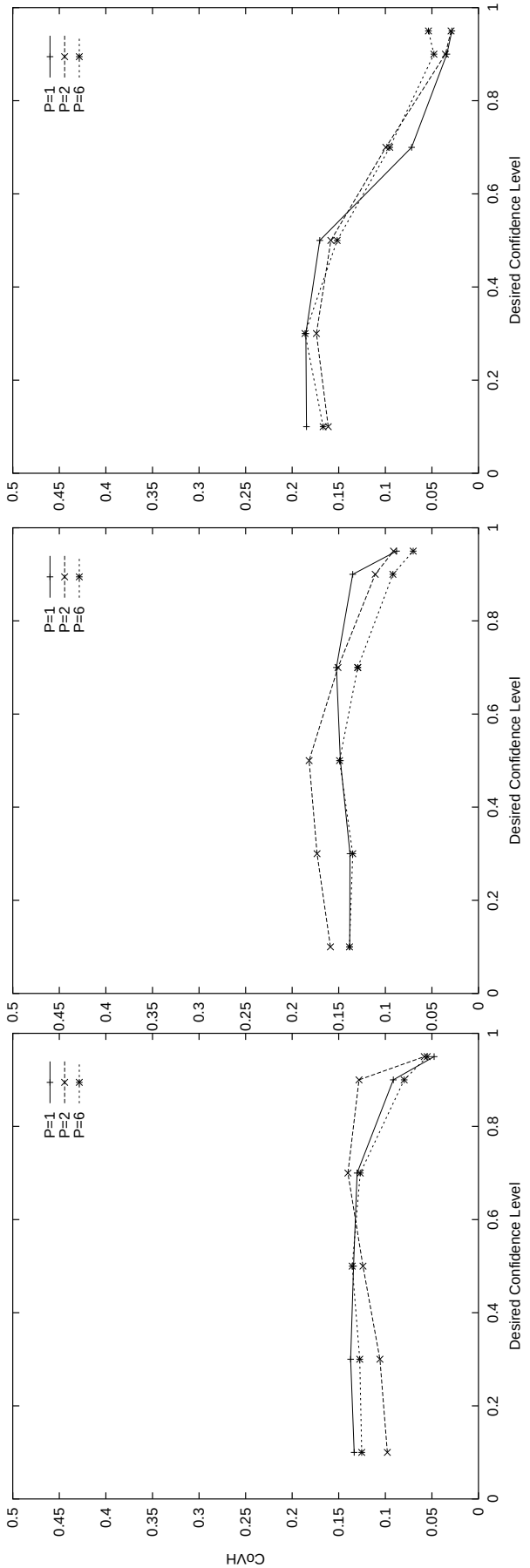


Fig. 63: SBM : CoV{H} for $M/D/1$ and $\rho = 0.1, 0.5, 0.9$

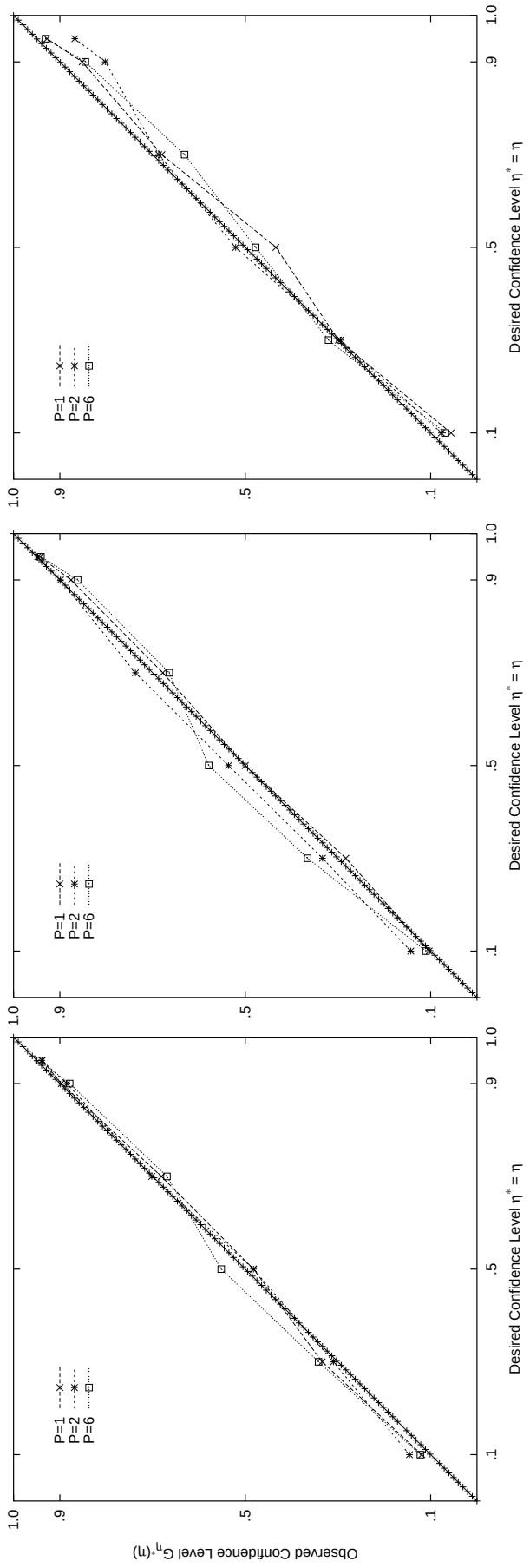


Fig. 64: SBM : coverage function for $M/E_4/1$ and $\rho = 0.1, 0.5, 0.9$

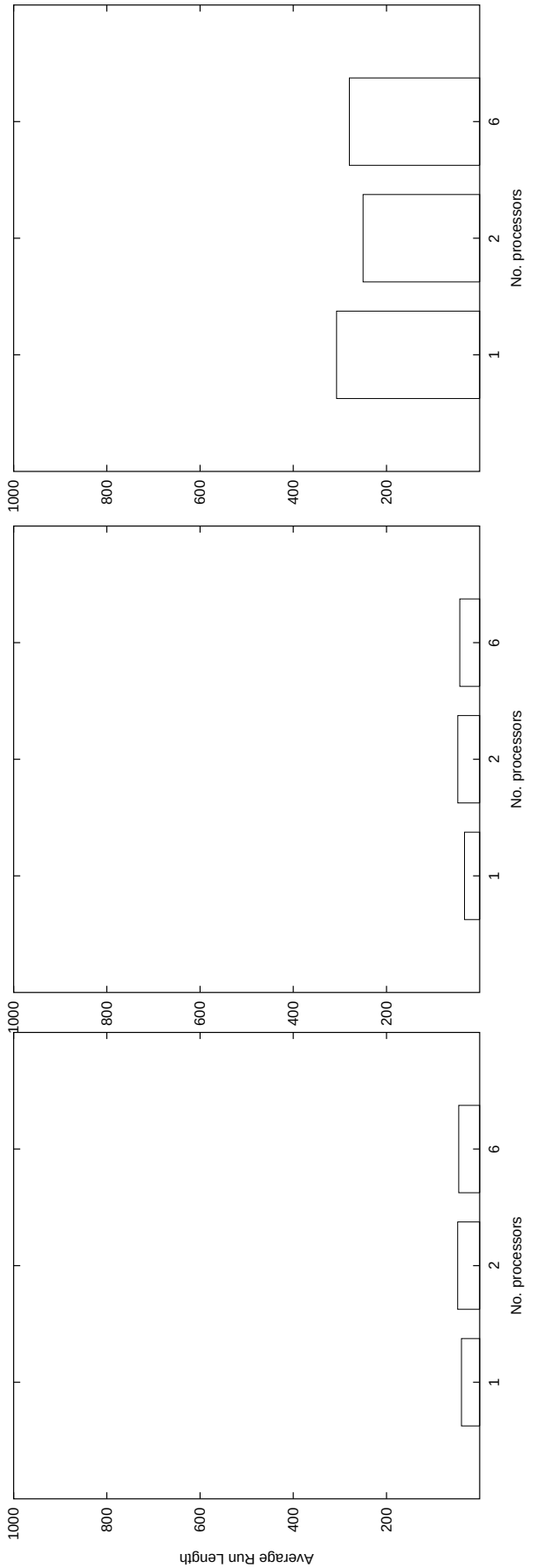


Fig. 65: SBM : average run length for $M/E_4/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

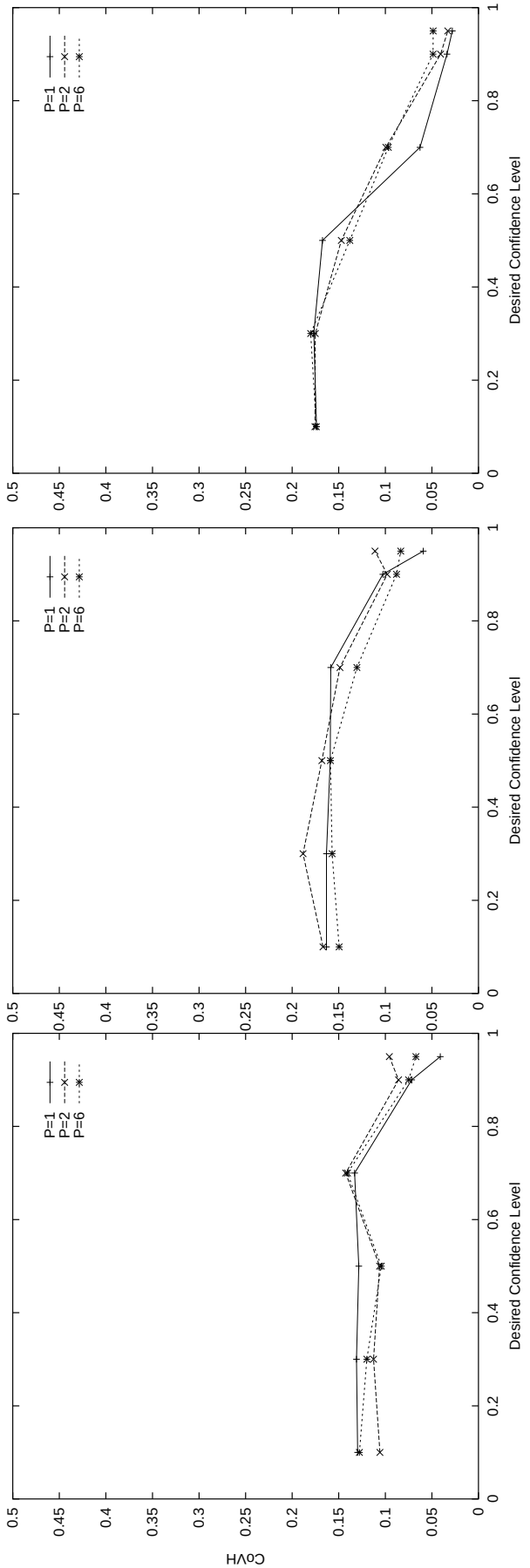


Fig. 66: SBM : CoV{H} for $M/E_4/1$ and $\rho = 0.1, 0.5, 0.9$

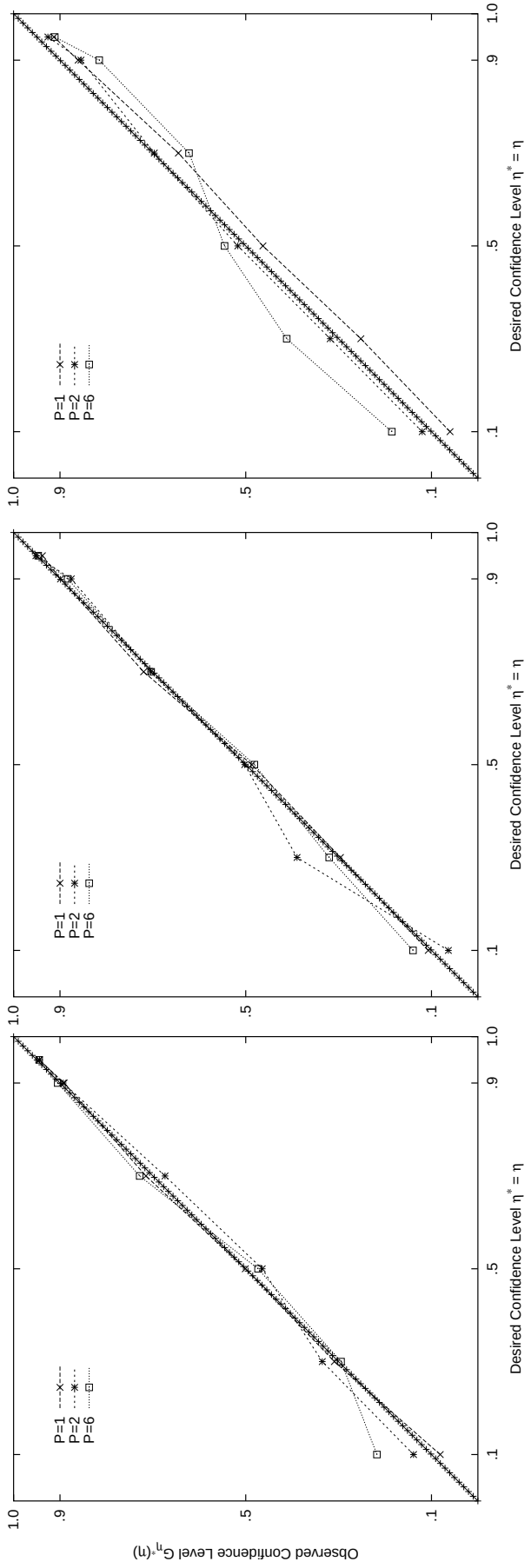


Fig. 67: SBM : coverage function for $M/M/1$ and $\rho = 0.1, 0.5, 0.9$

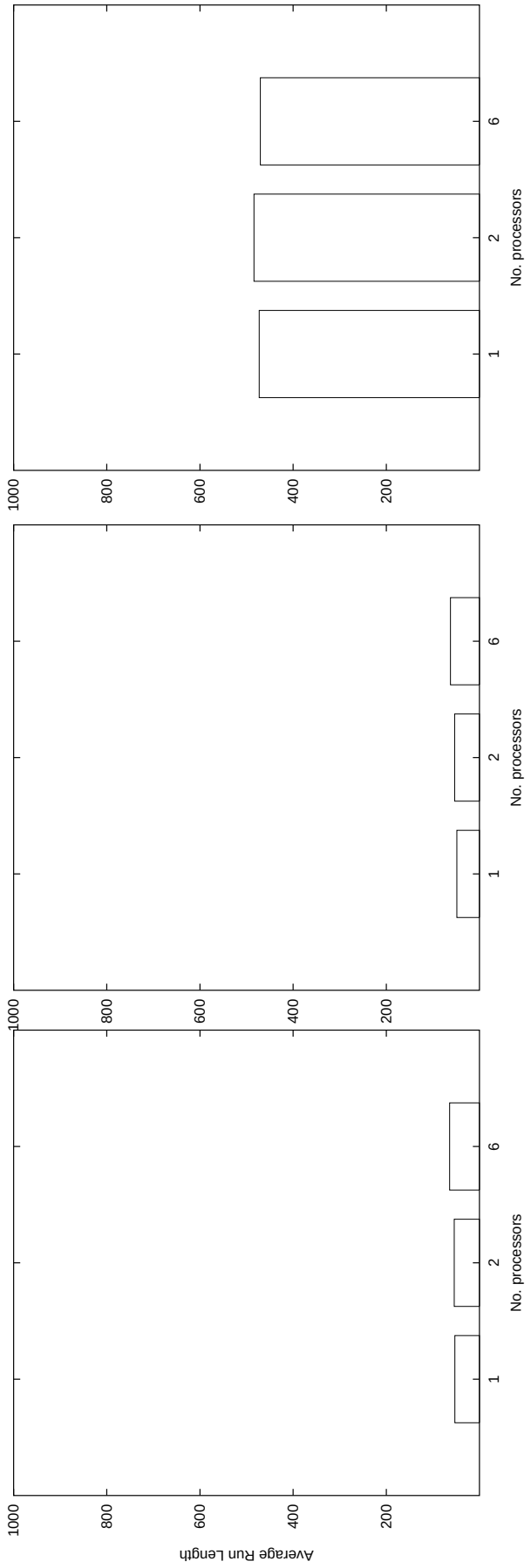


Fig. 68: SBM : average run length for $M/M/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

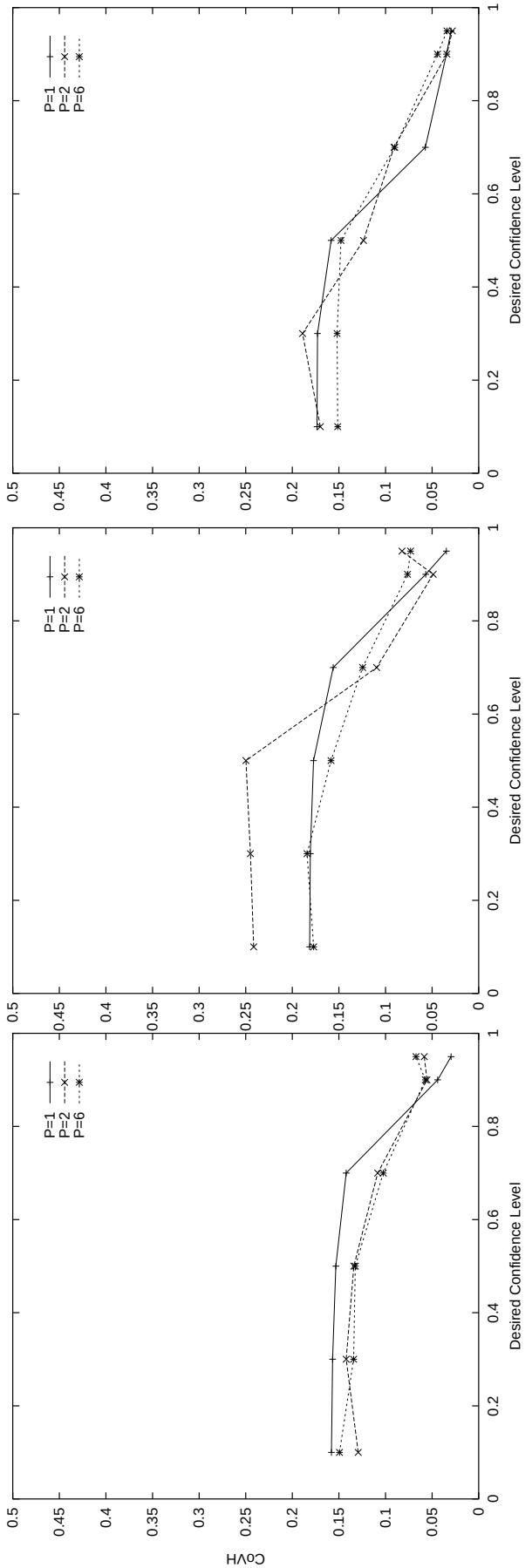


Fig. 69: SBM : CoV{H} for $M/M/1$ and $\rho = 0.1, 0.5, 0.9$

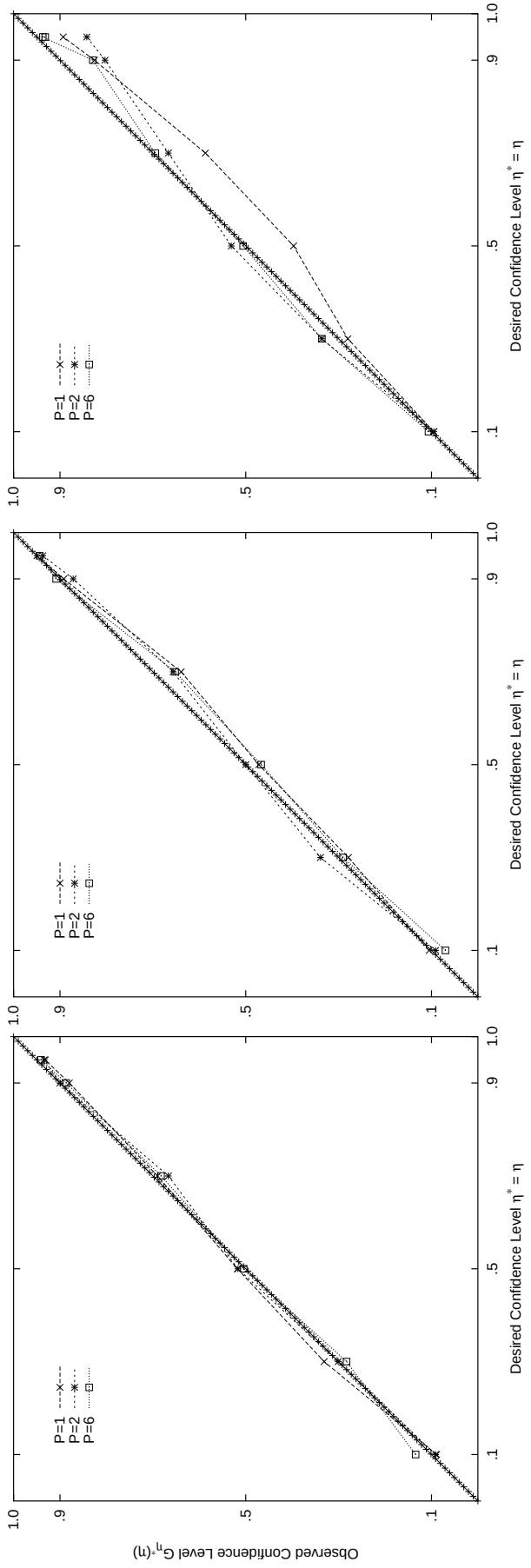


Fig. 70: SBM : coverage function for $M/H_2/1$ and $\rho = 0.1, 0.5, 0.9$

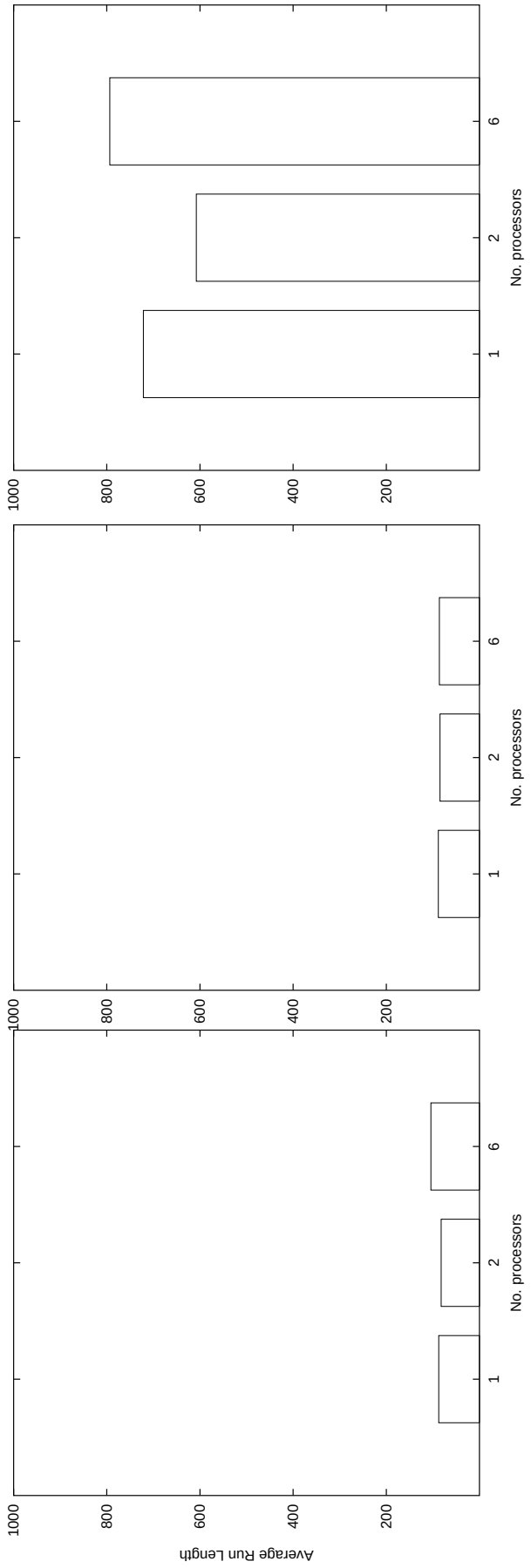


Fig. 71: SBM : average run length for $M/H_2/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

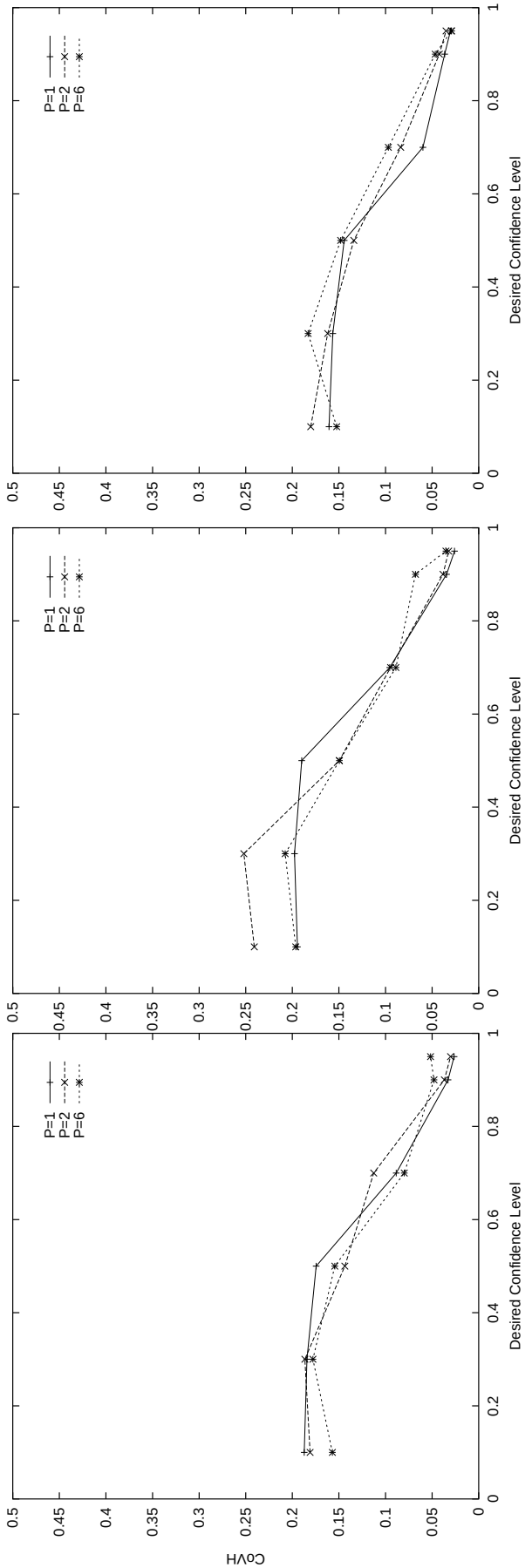


Fig. 72: SBM : CoV{H} for $M/H_2/1$ and $\rho = 0.1, 0.5, 0.9$

G

OBM PERFORMANCE

The following pages present the results of sequential coverage analysis by applying OBM, when simulating the average waiting time of the following queing systems : $M/D/1$, $M/E_4/1$, $M/M/1$ and $M/H_2/1$ (see App. C for more details of their settings). We varied traffic intensity ($\rho = 0.1, 0.5,$ and 0.9), confidence level ($\eta = 0.1$ to 0.95) and degree of parallelization (P = 1, 2 and 6).

We constructed a coverage function for each combination of these factors and, additionally, we assed the average run length (in units of 1000 observations) and CoV{H}, a measure of stability of confidence interval half-width. Graphics of the same type of MOE were plotted using the same unit, in such a way one can compare them visually. A few graphics corresponding to average run length of very low confidence level and traffic intensity, do not appear, since they used much less than 1000 observations.

The end of transient period was found by means of a stationarity test proposed by Schruben et al. [121], already implemented in Akaroa-2. Observations of transient period are discarded, and the stochastic process being simulated is considered to be a sample from a covariance stationary process.

Initially, batch size was selected to be equal 100 observations. Independence among batch means was pursued by using jackknife estimators of correlation coefficients. There was, initially, 100 batches, but after finding the optimal batch size that yielded almost uncorrelated batch means, observations were reorganized into 25 batches, according to [114], which resulted in a batch size four times larger.

At this moment, each observation initiated an overlapped batch, yielding (n - m + 1) batches. At consecutive checkpoints, sample mean and sample variance were calculated and sent to a global analyzer, together with the sample size.

The number of batches is increased by 2 whenever the relative precision is not achieved. Whenever the test for stopping rule fails, each new observation collected, together with the previous (m - 1) observations, completed a new overlapped batch. It corresponds to an increase of the number of batches. The simulation experiment stopped when the relative precision (5%) was achieved two times consecutively.

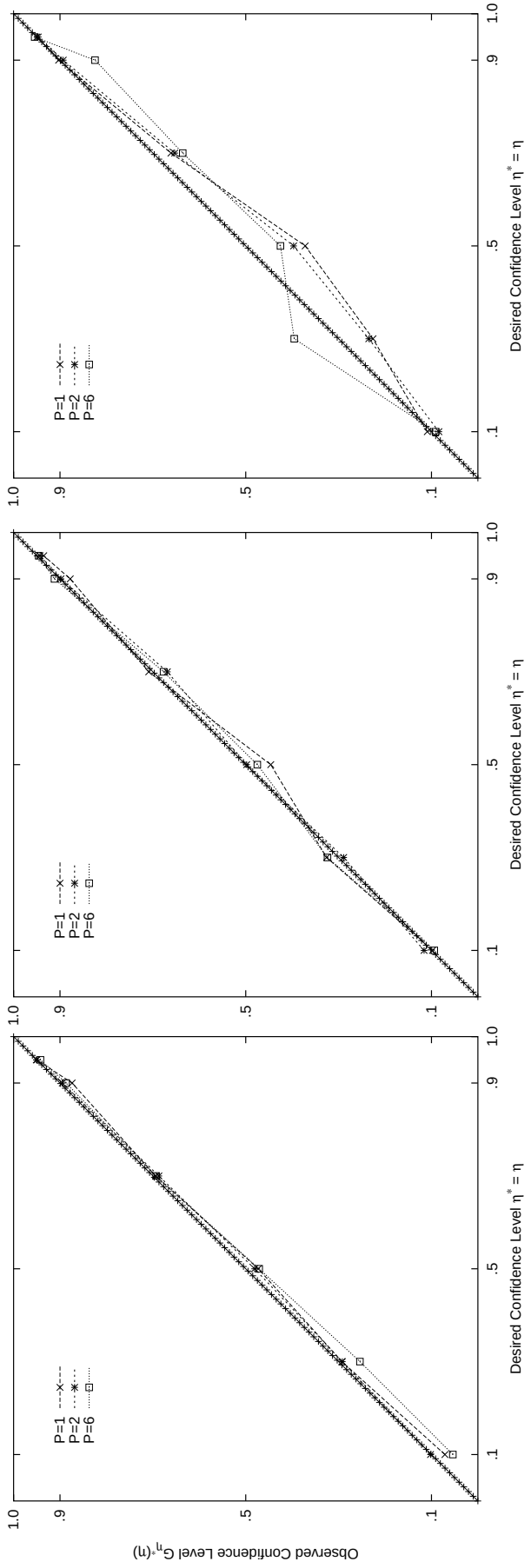


Fig. 73: OBM : coverage function for $M/D/1$ and $\rho = 0.1, 0.5, 0.9$

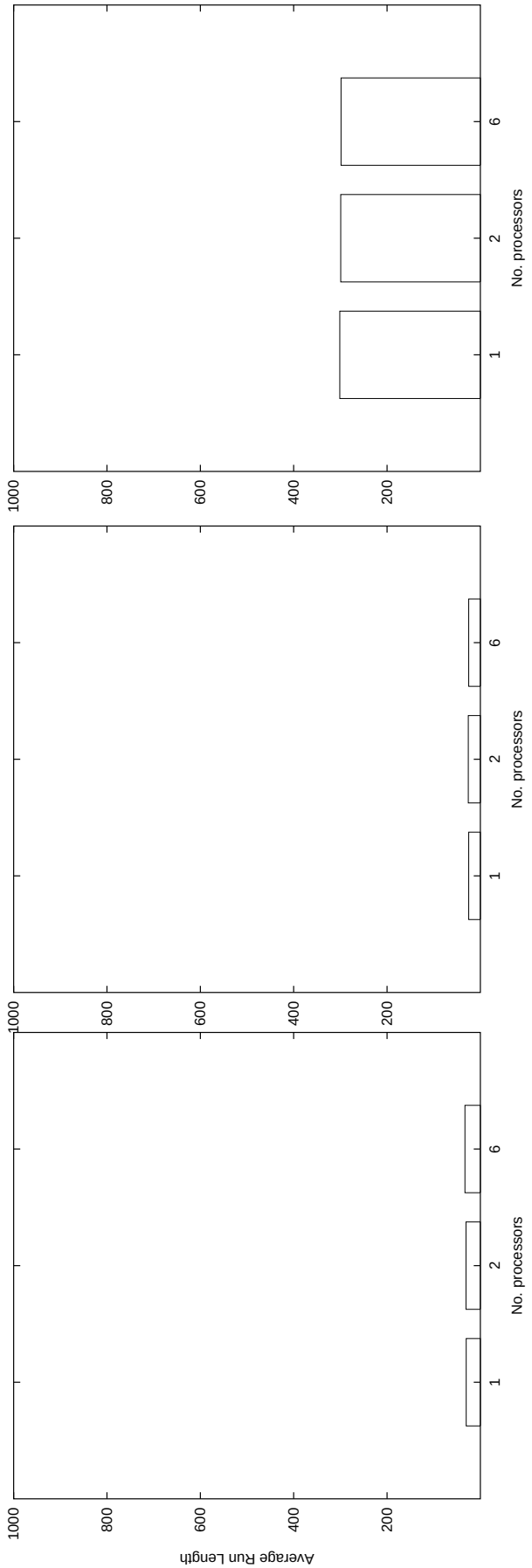


Fig. 74: OBM : average run length for $M/D/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

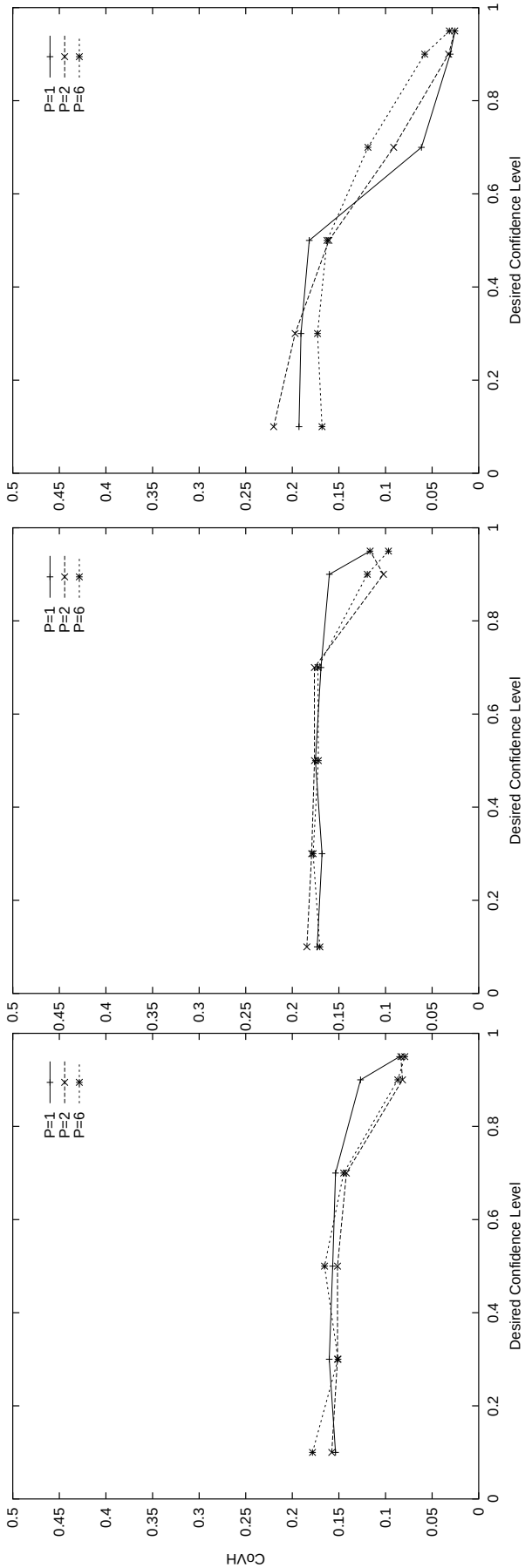


Fig. 75: OBM : CoV{H} for $M/D/1$ and $\rho = 0.1, 0.5, 0.9$

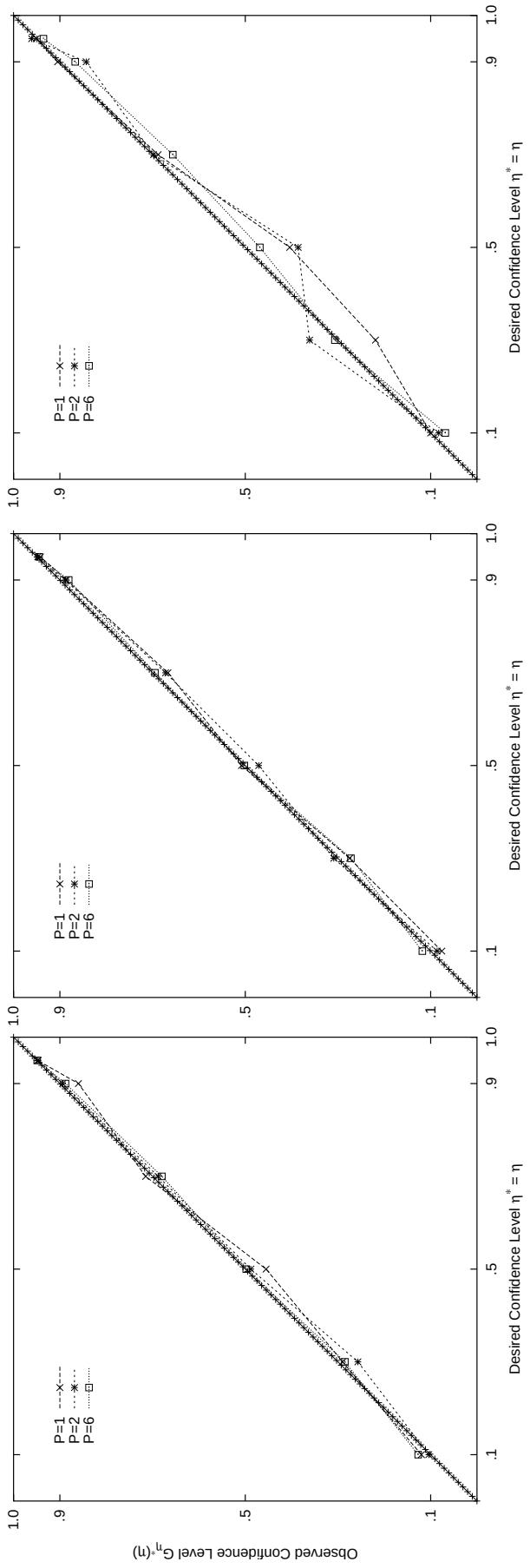


Fig. 76: OBM : coverage function for $M/E_4/1$ and $\rho = 0.1, 0.5, 0.9$

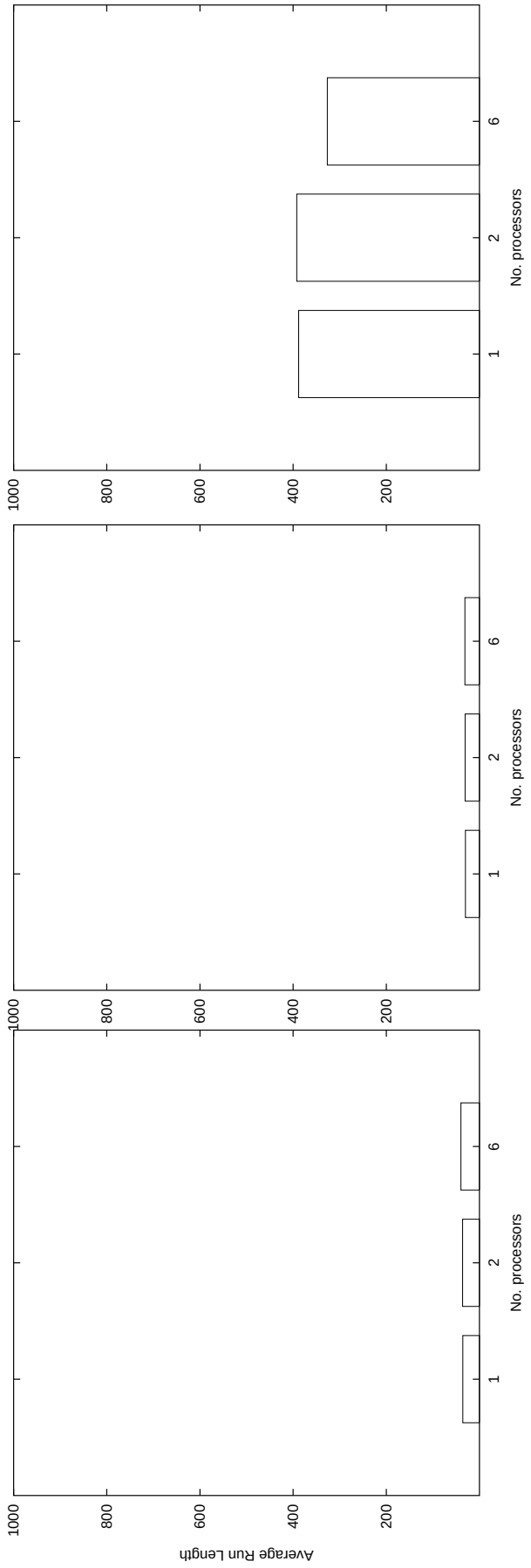


Fig. 77: OBM : average run length for $M/E_4/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

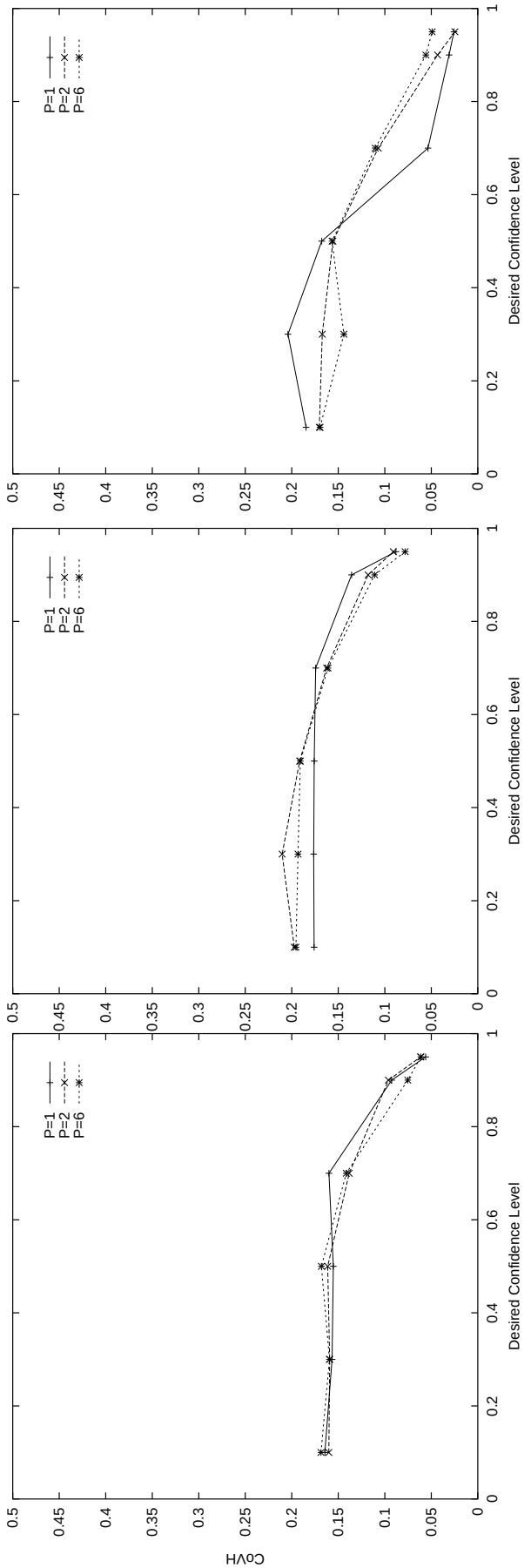


Fig. 78: OBM : CoV{H} for $M/E_4/1$ and $\rho = 0.1, 0.5, 0.9$

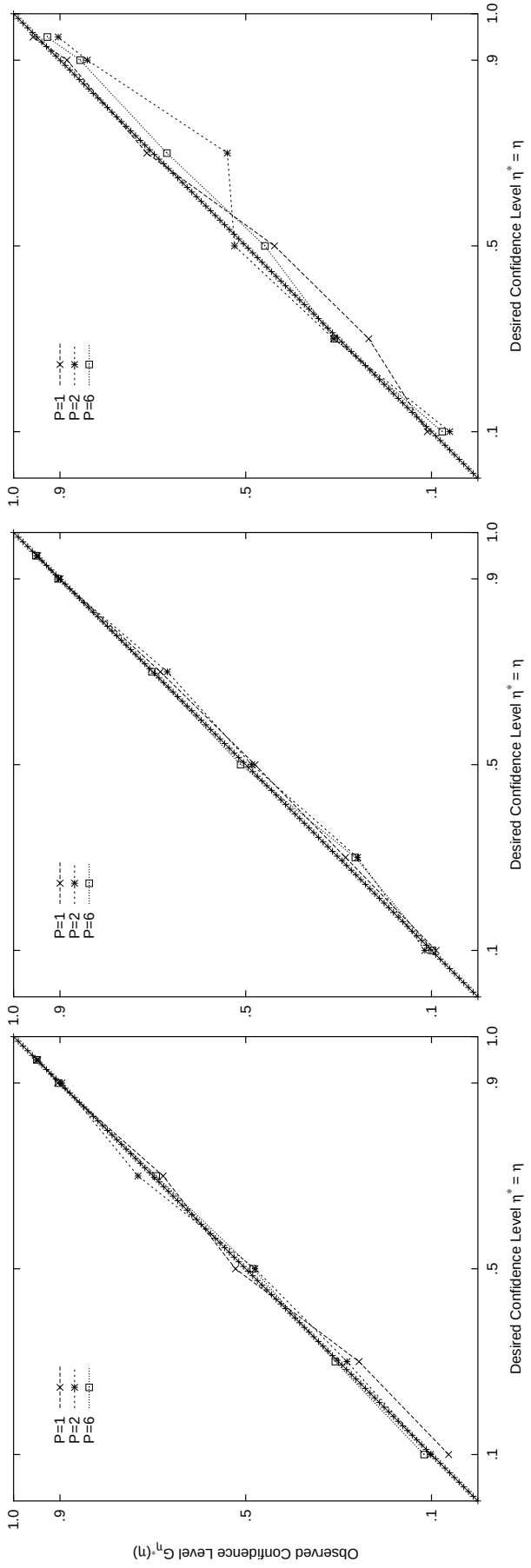


Fig. 79: OBM : coverage function for $M/M/1$ and $\rho = 0.1, 0.5, 0.9$

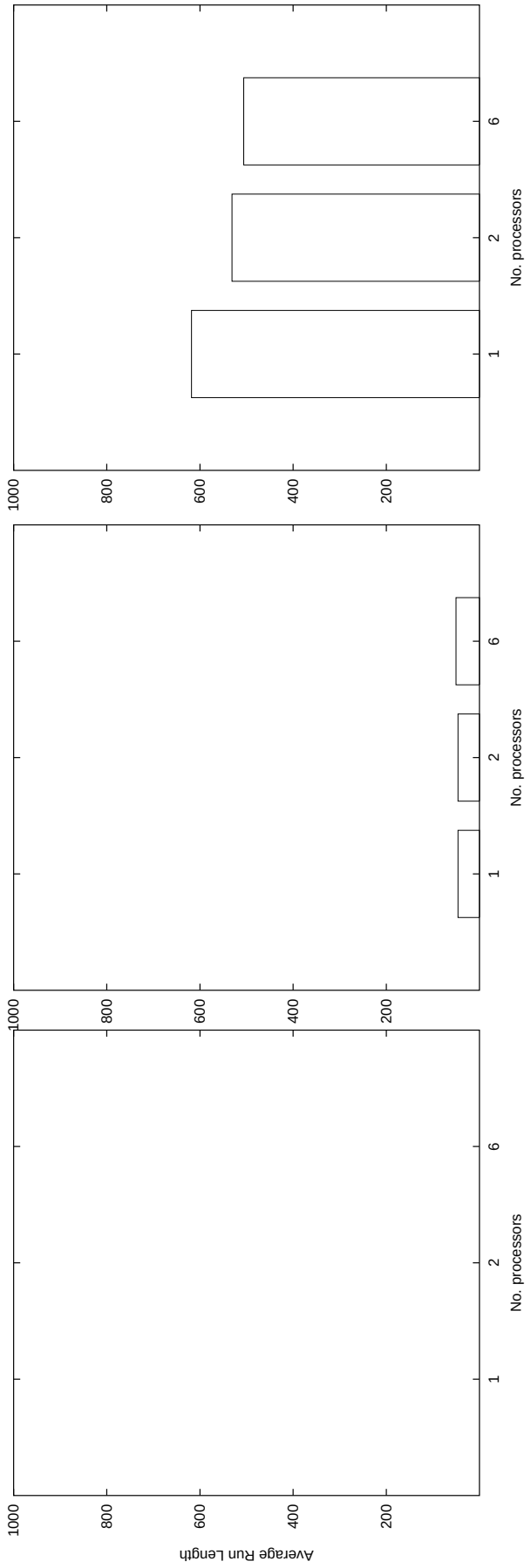


Fig. 80: OBM : average run length for $M/M/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

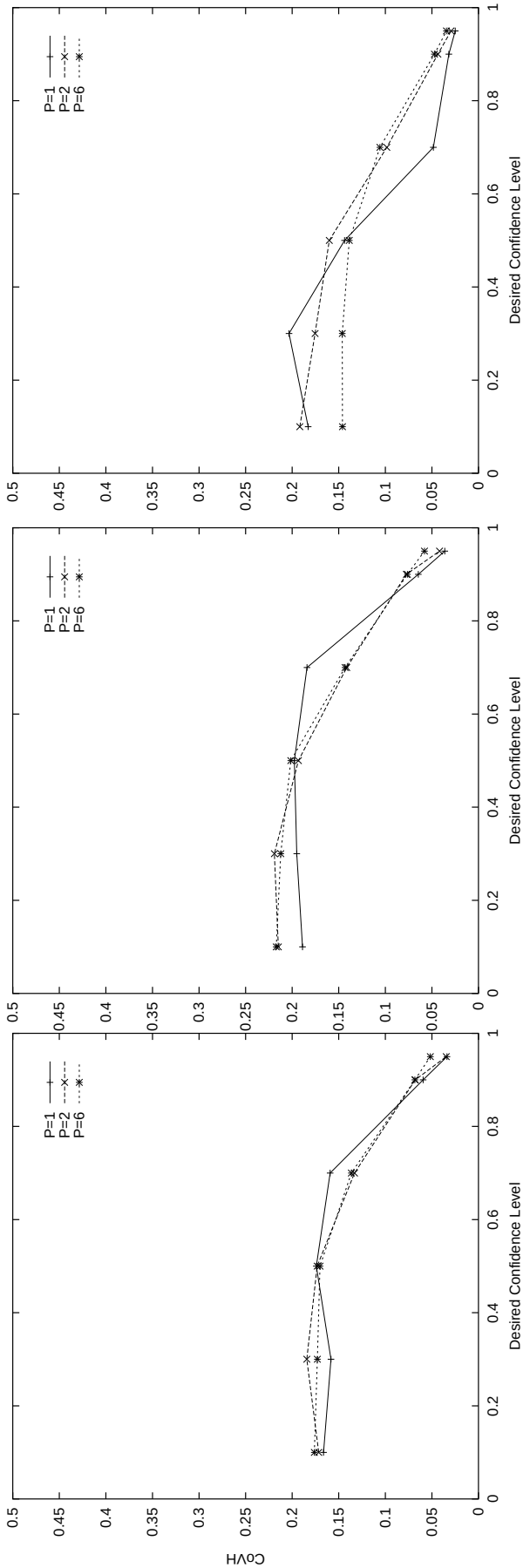


Fig. 81: OBM : CoV{H} for $M/M/1$ and $\rho = 0.1, 0.5, 0.9$

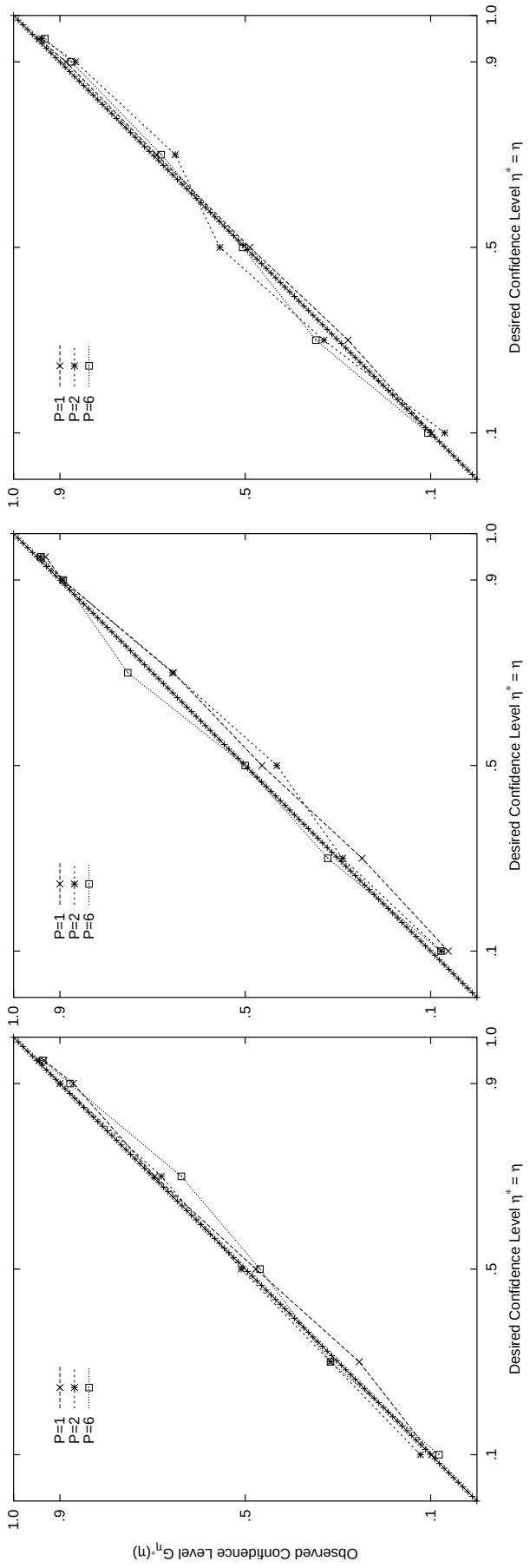


Fig. 82: OBM : coverage function for $M/H_2/1$ and $\rho = 0.1, 0.5, 0.9$

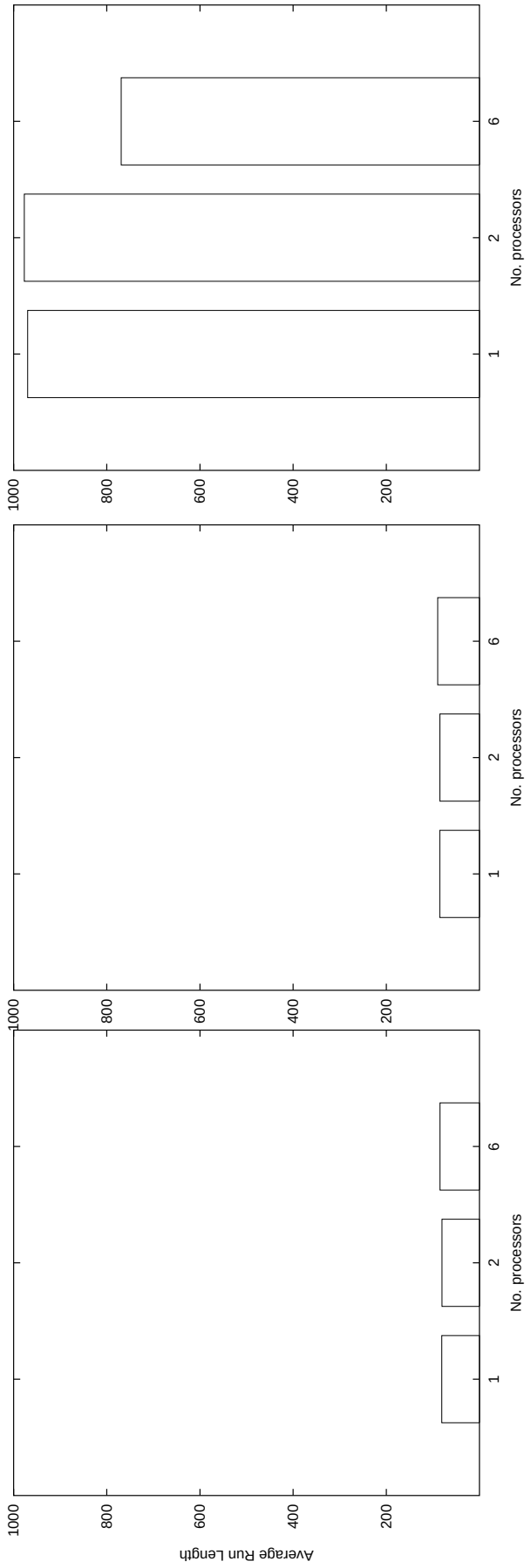


Fig. 83: OBM : average run length for $M/H_2/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

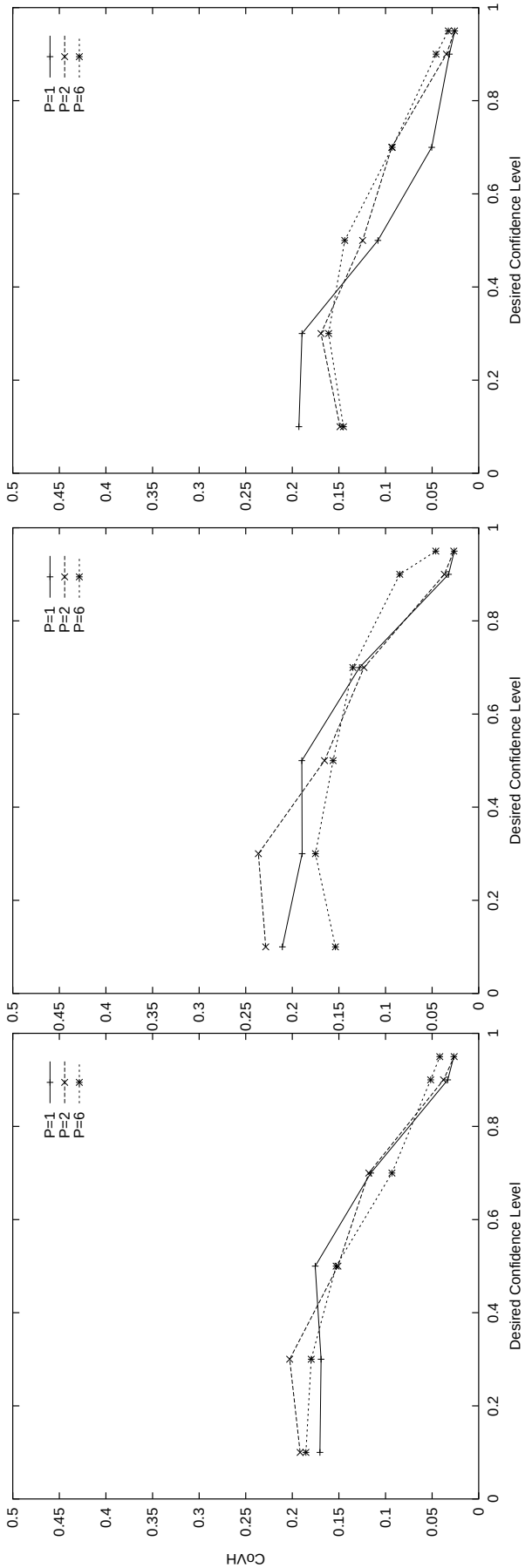


Fig. 84: OBM : CoV{H} for $M/H_2/1$ and $\rho = 0.1, 0.5, 0.9$

H

CSUM PERFORMANCE

The following pages present the results of sequential coverage analysis by applying STS/CSUM, when simulating the average waiting time of the following queing systems : $M/D/1$, $M/E_4/1$, $M/M/1$ and $M/H_2/1$ (see App. C for more details of their settings). We varied traffic intensity ($\rho = 0.1, 0.5$, and 0.9), confidence level ($\eta = 0.1$ to 0.95) and degree of parallelization (P = 1, 2 and 6).

We constructed a coverage function for each combination of these factors and, additionally, we assed the average run length (in units of 1000 observations) and CoV{H}, a measure of stability of confidence interval half-width. Graphics of the same type of MOE were plotted using the same unit, in such a way one can compare them visually. A few graphics corresponding to average run length of very low confidence level and traffic intensity, do not appear, since they used much less than 1000 observations.

The end of transient period was found by means of a stationarity test proposed by Schruben et al. [121], already implemented in Akaroa-2. Observations of transient period are discarded, and the stochastic process being simulated is considered to be a sample from a covariance stationary process.

Initially, batch size was selected to be equal 100 observations. A random variable A_i were computed to the ith batch, according to Eq. 2.21. Normality among A_i's was pursued by using the Shapiro-Wilk test (look at App. A for more details). We decided to work with 10 batches in order to keep in touch with the degrees of freedom suggested in [114].

Finding the normality among A_i's, at consecutive checkpoints, sample mean and sample variance were calculated and sent to a global analyzer, together with the sample size. The number of batches is kept fixed. The simulation experiment stopped when the relative precision (5%) was achieved two times consecutively.

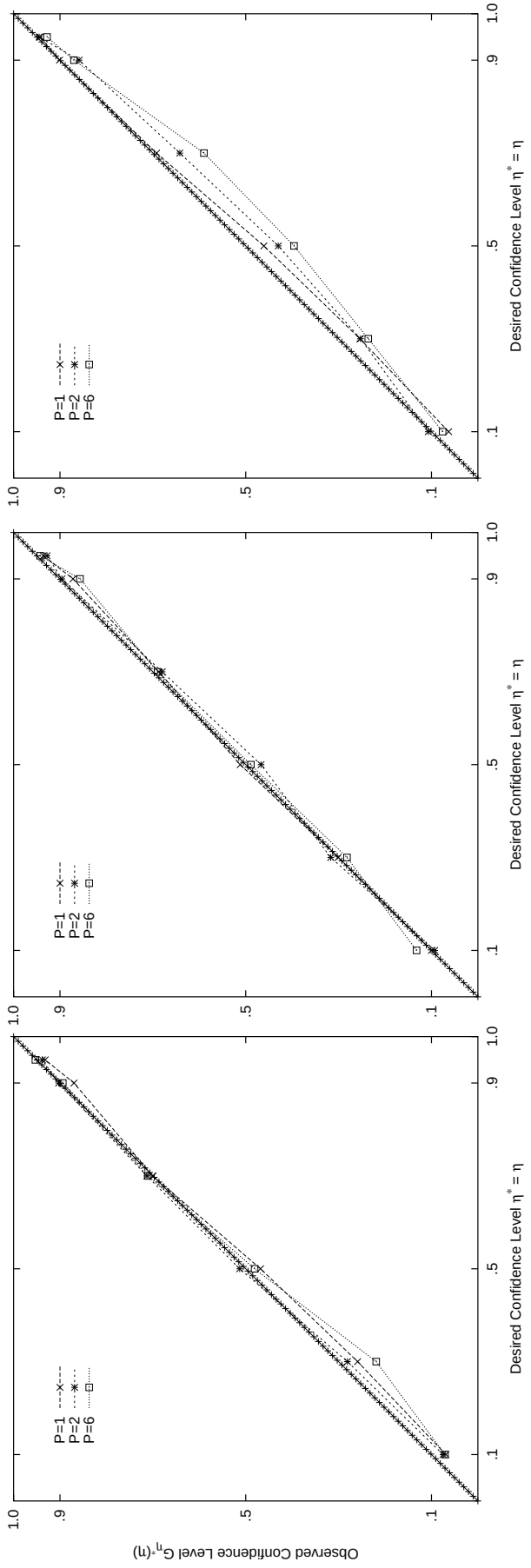


Fig. 85: CSUM : coverage function for $M/D/1$ and $\rho = 0.1, 0.5, 0.9$

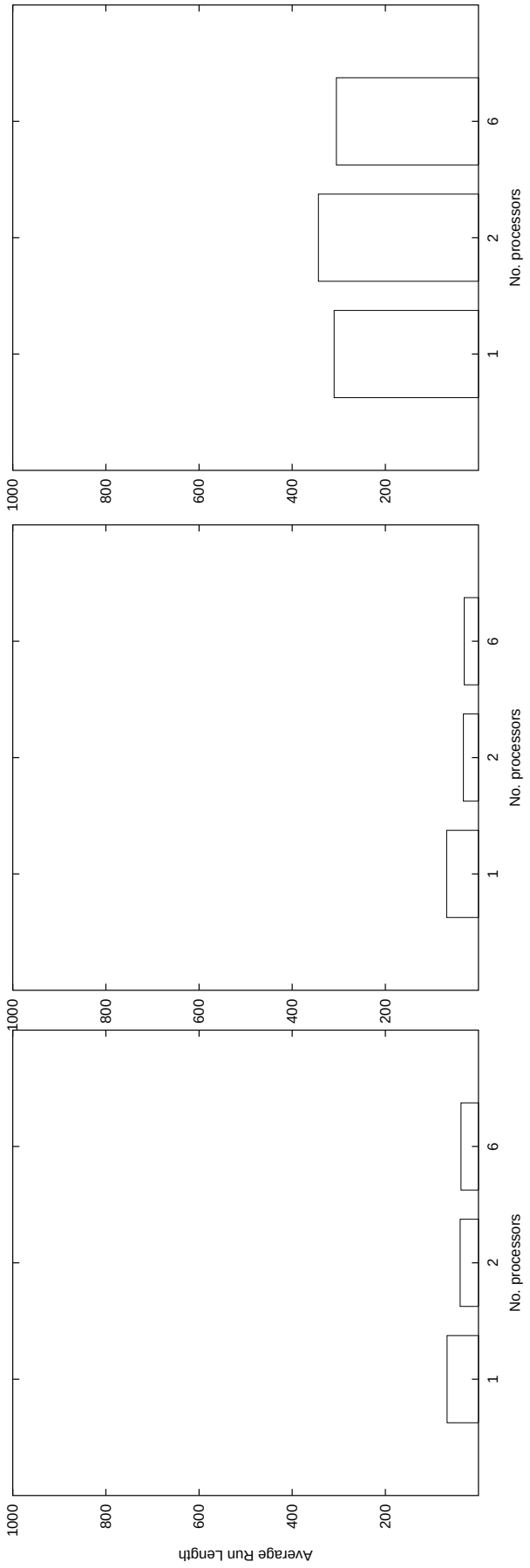


Fig. 86: CSUM : average run length for $M/D/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

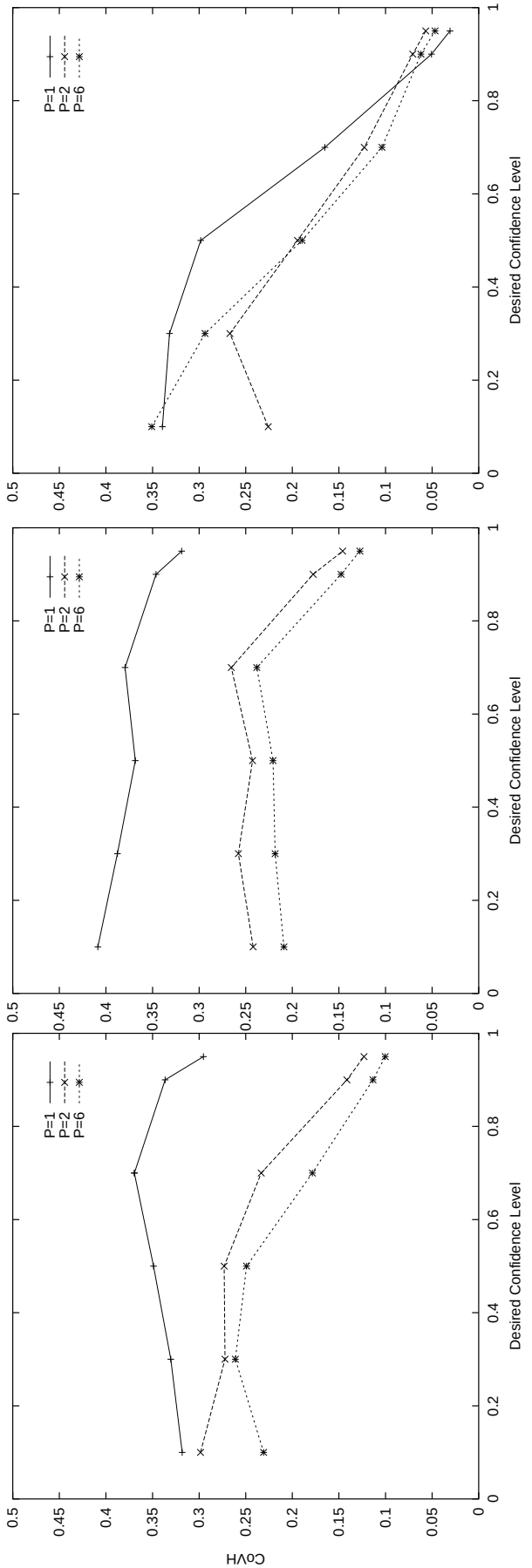


Fig. 87: CSUM : CoV{H} for $M/D/1$ and $\rho = 0.1, 0.5, 0.9$

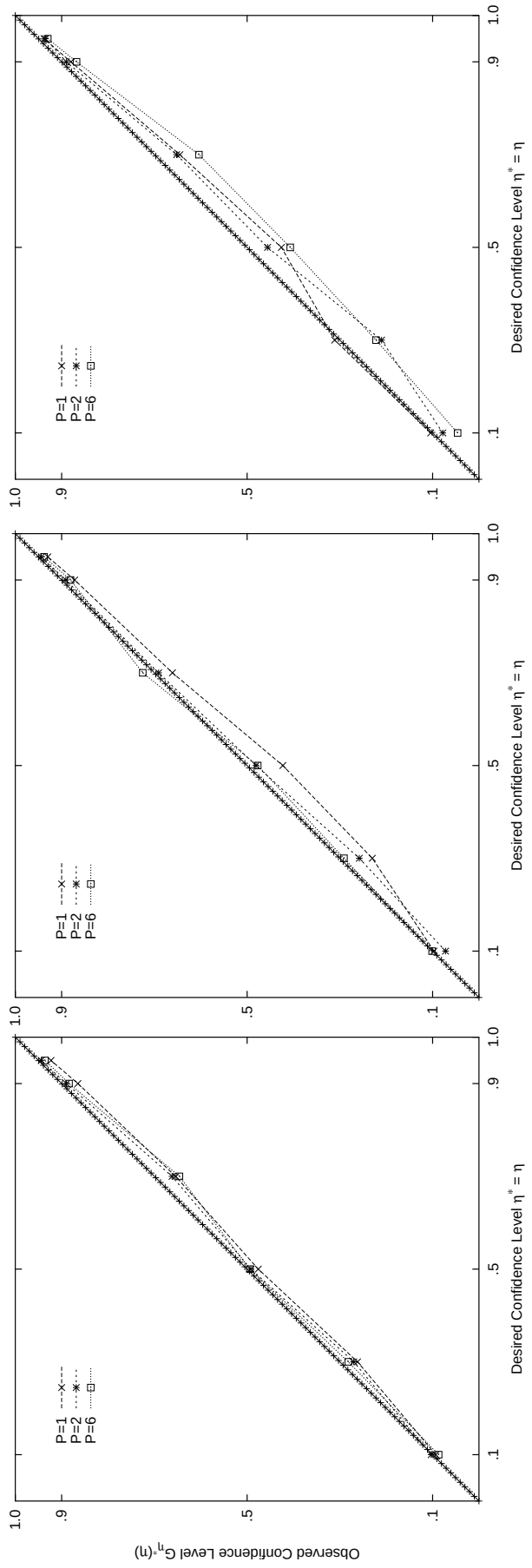


Fig. 88: CSUM : coverage function for $M/E_4/1$ and $\rho = 0.1, 0.5, 0.9$

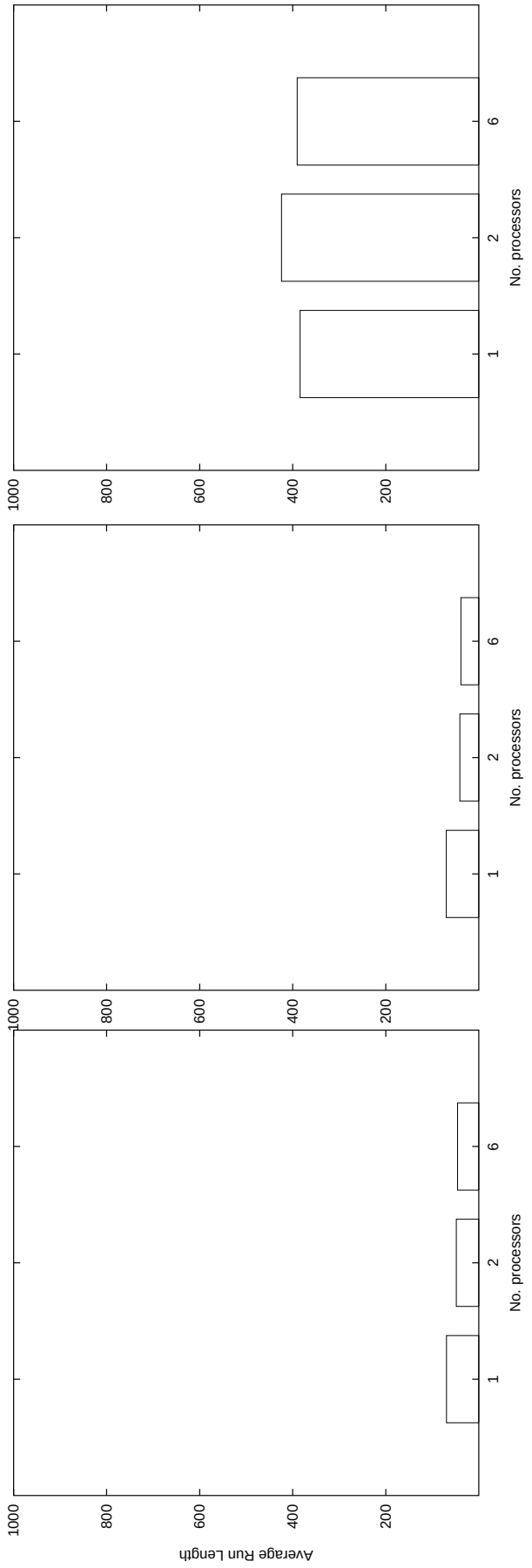


Fig. 89: CSUM : average run length for $M/E_4/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

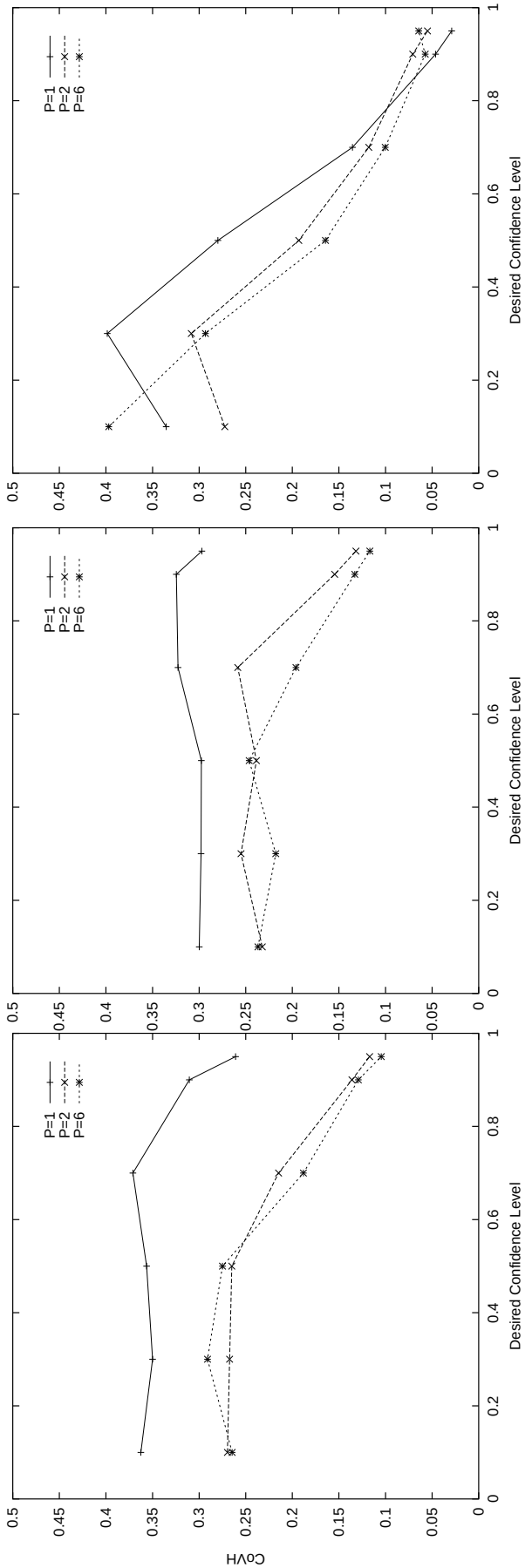


Fig. 90: CSUM : CoV{H} for $M/E_4/1$ and $\rho = 0.1, 0.5, 0.9$

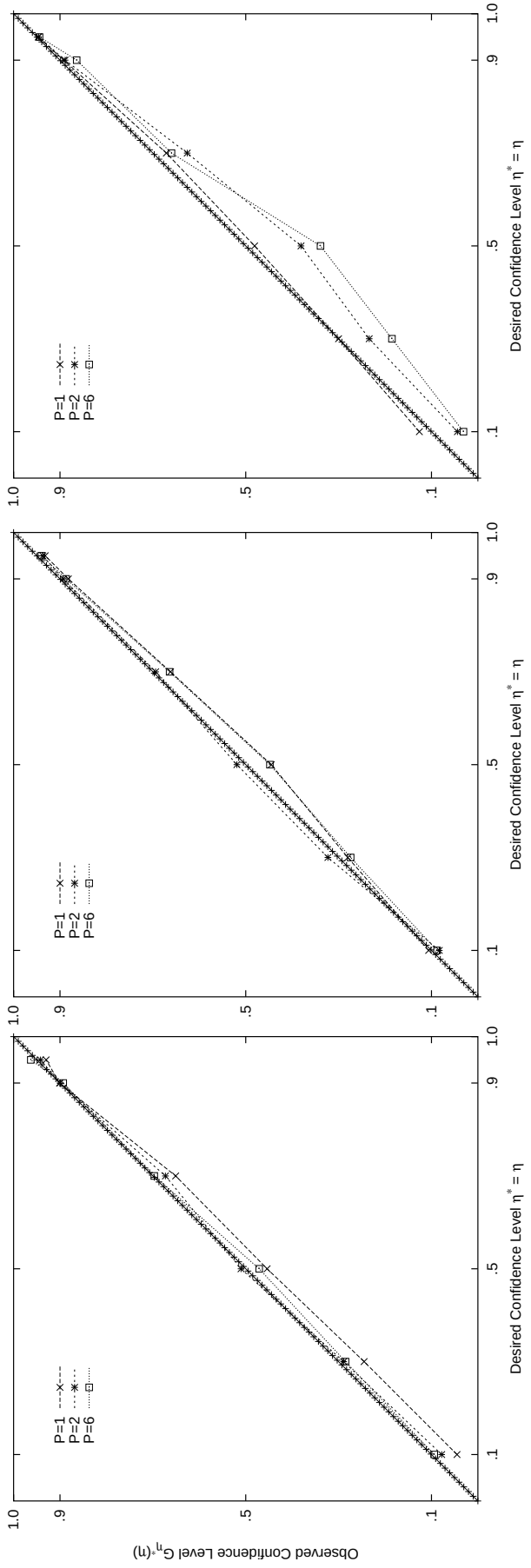


Fig. 91: CSUM : coverage function for $M/M/1$ and $\rho = 0.1, 0.5, 0.9$

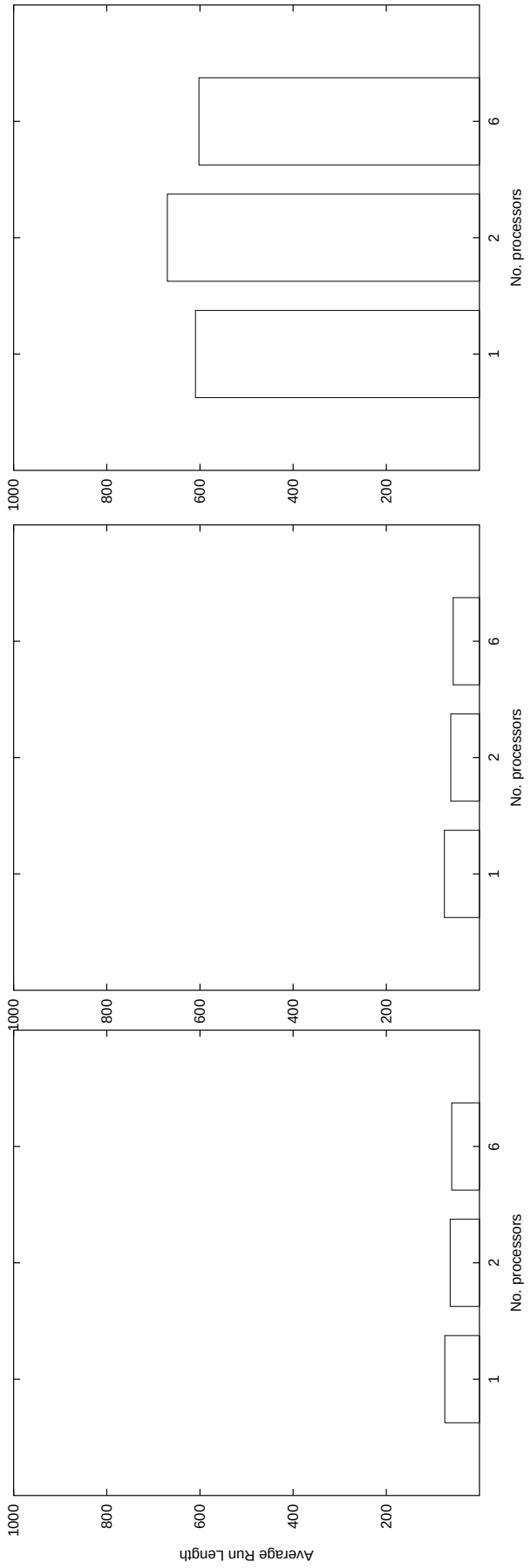


Fig. 92: CSUM : average run length for $M/M/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

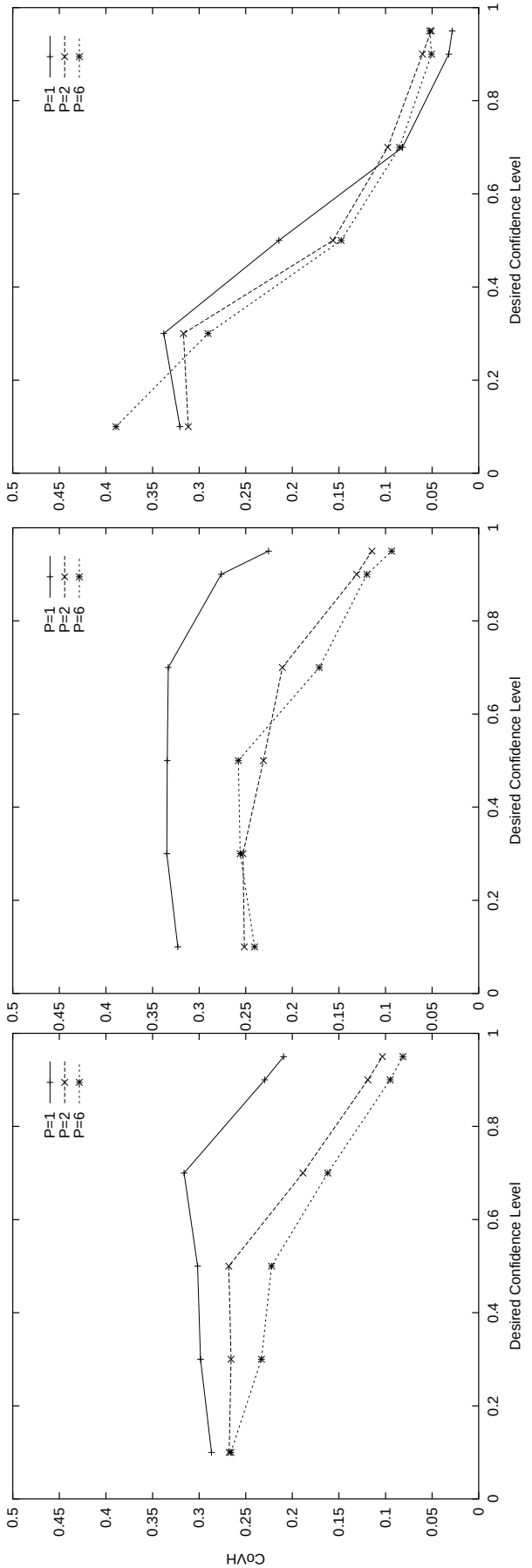


Fig. 93: CSUM : CoV{H} for $M/M/1$ and $\rho = 0.1, 0.5, 0.9$

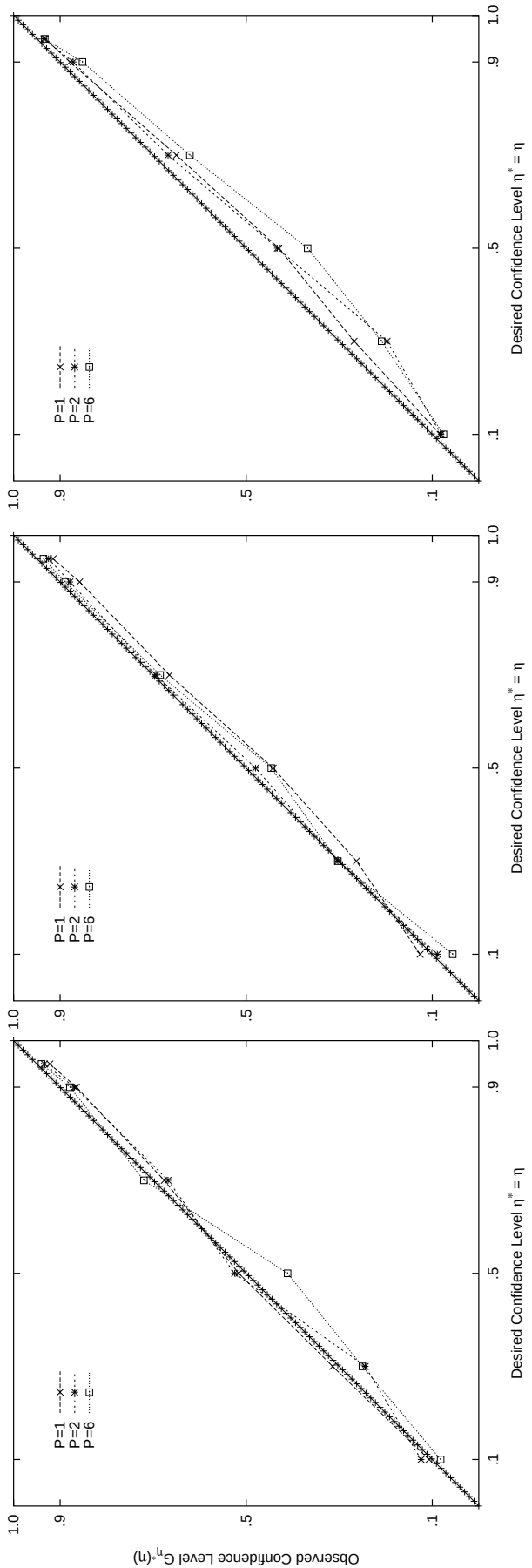


Fig. 94: CSUM : coverage function for $M/H_2/1$ and $\rho = 0.1, 0.5, 0.9$

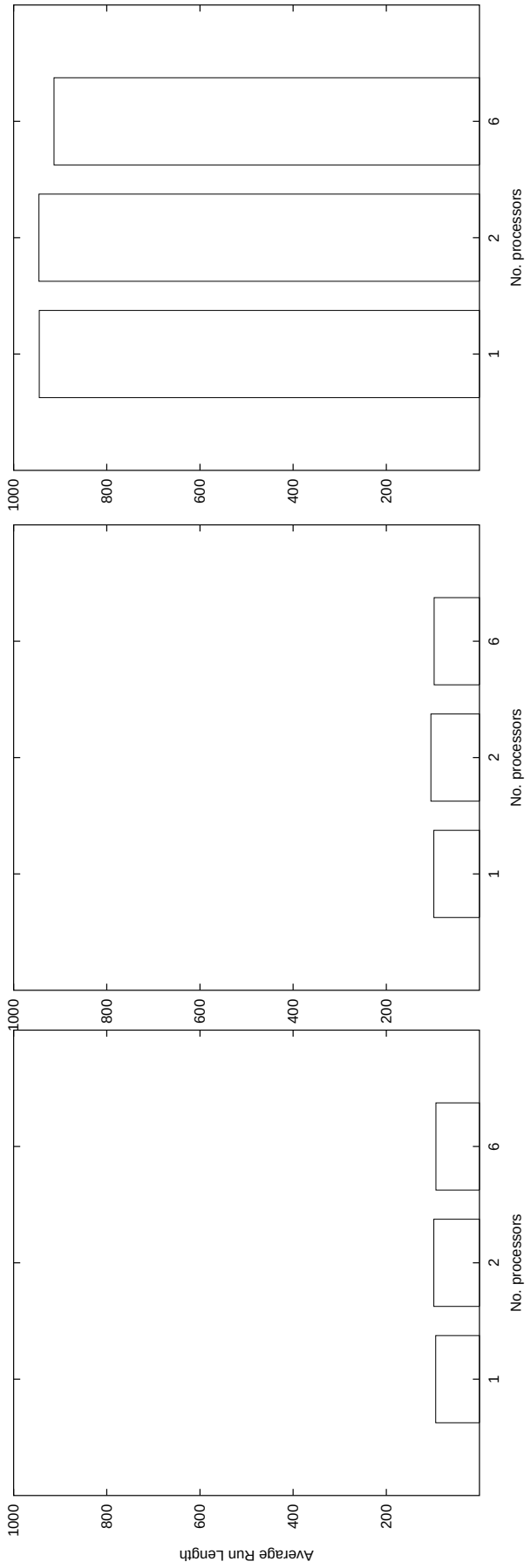


Fig. 95: CSUM : average run length for $M/H_2/1$, $\eta = 0.95$ and $\rho = 0.1, 0.5, 0.9$

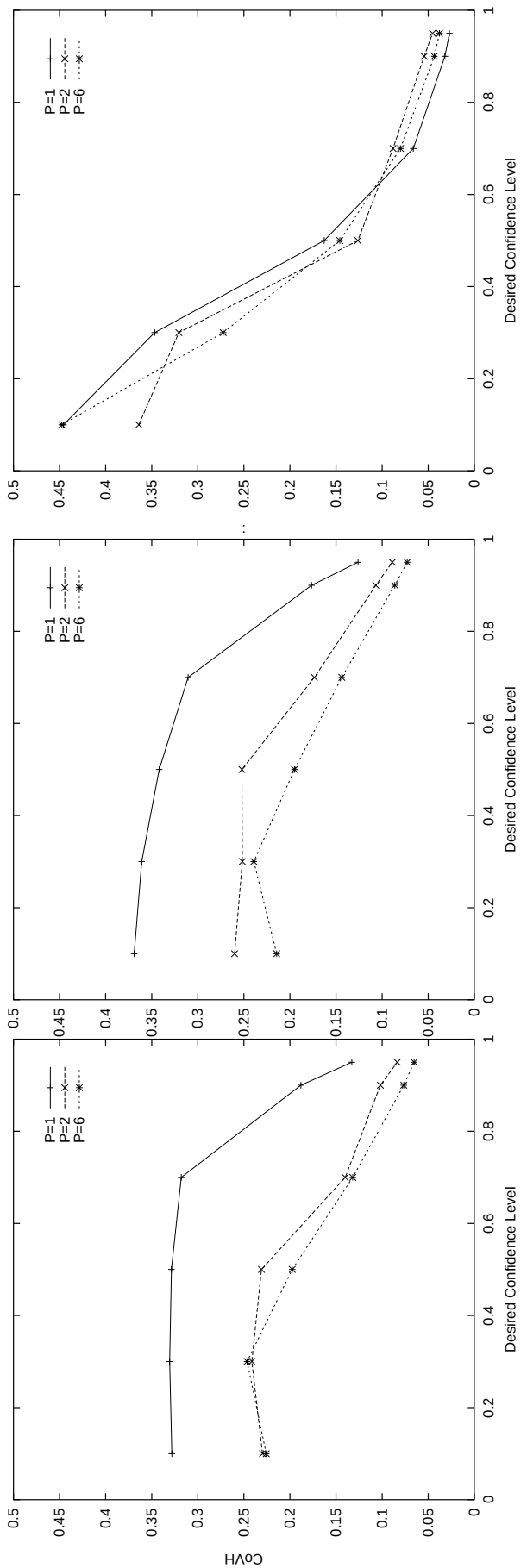


Fig. 96: CSUM : CoV{H} for $M/H_2/1$ and $\rho = 0.1, 0.5, 0.9$

References

[1] Adam, N. Achieving a confidence interval for parameters estimated by simulation. *Management Science, Nr. 7, July 29* (1983), 856–866.

[2] Alexopoulos, C., Fishman, C., and Seila, A. Computational experience with the batch means method. In *Winter Simulation Conference* (New York, Dec. 1997), ACM Association for Computing Machinery, pp. 194–201.

[3] Alexopoulos, C., and Seila, A. Implementing the batch means method in simulation experiments. In *Winter Simulation Conference* (New York, Dec. 1996), ACM Association for Computing Machinery, pp. 214–221.

[4] Amdahl, G. Validity of the single procesor approach to achieving large scale computing capabilities. In *AFIPS Conference Proceedings* (1976), pp. 483–485.

[5] Anderson, O. *The Statistical Analysis of Time Series*. John Willey, 1971.

[6] Andradóttis, S. Variance estimation using replicated batch means. In *Winter Simulation Conference* (New York, Dec. 2001), ACM Association for Computing Machinery, pp. 338–343.

[7] Andreassen, R., Heegaard, P., and Helvik, B. Importance sampling for speed-up simulation of heterogeneous mpeg sources. pp. 190–203.

[8] Añonuevo, R., and Nelson, B. Automated estimation and variance reduction via control variates for infinite-horizon simulations. *Comput. Opns. Res 15(5)* (1988), 447–456.

[9] Asgarkhani, M., and Pawlikowski, K. Simulation studies of mixed traffic on satellite channels using tdma-reservation protocol. In *Proceedings of the 8^{th} Phoenix International Conference on Computers and Communications* (1989), IEEE, pp. 195–200.

[10] Ayani, R., Ismailov, A., P., Rajaei, H., and Rönngren. Modeling and simulation of a high-speed lan. *Simulation 64* (1995), 7–14.

[11] Bartels, R. The rank version of von neumann's ratio test for randomness. *Journal American Statistical Association 77*, 377 (1982), 40–46.

[12] Bertsekas, D., and Gallager, R. *Data Networks*, 2nd ed. Prentice Hall, 1992.

[13] Bhavsar, V., and Isaac, J. Design and analysis of parallel Monte Carlo algorithms. *SIAM Journal on Scientific and Statistical Computing 8*, 1 (Jan. 1987), S73–S95.

[14] Billingsley, P. *Convergence of Probability Measures*. John Wiley, New York, 1999.

[15] Bischak, D., Kelton, W.D., and Pollock, S. Weighted batch means for confidence intervals on steady-state simulations. *Management Science, Nr. 8, August 39* (1993), 1002–1019.

[16] Box, G., and Jenkins, G. *Time Series Analysis Forecasting and Control*. Holden-Day, San Francisco, 1970.

[17] Brillinger, D. Estimation of the mean of a stationary time series by sampling. *Journal of Applied Probability 10, 2* (1973), 419–431.

[18] Charnes, J., Carson, J., Dewsnup, M., Seila, A., Tew, J., and Sadowski, R. Output analysis research: Why bother ? In *Winter Simulation Conference* (New York, Dec. 1994), ACM Association for Computing Machinery, pp. 399–405.

[19] Chen, C., and Kelton, W. A stopping procedure based on phi-mixing conditions. In *Winter Simulation Conference* (New York, Dec. 2000), ACM Association for Computing Machinery, pp. 617–623.

[20] Cheng-Chang, S., Heidelberger, P., and Shahabuddin, P. FAST SIMULATION OF PACKET LOSS RATES IN A SHARED BUFFER COMMUNICATIONS SWITCH. Technical Report TR-93-79, Institute for Computer Applications in Science and Engineering, Nov. 1993.

[21] Chien, C. Batch size selection for the batch means method. In *Winter Simulation Conference* (New York, Dec. 1994), ACM Association for Computing Machinery, pp. 345–352.

[22] Chow, Y., and Robbins, H. On the asymptotic theory of fixed-width sequential confidence intervals for the mean. *Ann. Math. Stat. 36* (1965), 457–462.

[23] Conway, R. Some tactical problems in digital simulation. *Manag. Sci. 10, 1* (1963), 47–61.

[24] Crane, M., and Iglehart, D. Simulating stable stochastic systems, I: General multiserver queues. *Journal of the ACM 21*, 1 (Jan. 1974), 103–113.

[25] Crane, M., and Iglehart, D. Simulating stable stochastic systems, II: Markov chains. *Journal of the ACM 21*, 1 (Jan. 1974), 114–123.

[26] Crane, M., and Iglehart, D. Simulating stable stochastic systems, III: Regenerative processes and discrete-event simulations. *Operations Research 23, 1* (1975), 33–45.

[27] D'Agostino, R., and Stephens, M. Goodnessof -fit techniques, 1986.

[28] Devetsikiotis, M., and Townsend, J. Statistical optimization of dynamic importance sampling parameters for efficient simulation of communication networks. *IEEE/ACM Transactions on Networking 1, 3* (1993), 293–305.

[29] Duersch, R., and Schruben, L. An interactive run length control for simulations on pcs. In *Winter Simulation Conference* (New York, Dec. 1986), ACM Association for Computing Machinery, pp. 866–869.

[30] Eager, D., Zahorjan, J., and Lazowska, E. Speedup vs efficiency in parallel systems. *IEEE Transactions on Computers 38*, 3 (Mar. 1989), 408–423.

[31] Efron, B., and Tibshirani, R. J. *An Introduction to the Bootstrap.* Chapman & Hall, New York, 1993.

[32] Ewing, G., McNickle, M., and Pawlikowski, K. Multiple replications in parallel: Distributed generation of data for speedingup quantitative. In *15th IMACS* (Aug. 1997), pp. 397–402.

[33] Ewing, G., Pawlikowski, K., and McNickle, K. Akaroa-2 : Exploiting network computing for distributing stochastic simulation. *Proceedings 13th European Simulation Multiconference* (June 1999), 129–134.

[34] Feller, W. *An Introduction to Probability Theory and Its Applications, volume II, Second edition.* 1971.

[35] Fishman, G. Problems in the statistical analysis of simulation experiments. *Communications of the ACM 10*, 2 (Feb. 1967).

[36] Fishman, G. Estimating sample size in computer simulation experiments. *Management Science, Nr. 8* (1971), 21–38.

[37] Fishman, G. Statistical analysis for queueing simulations. *Management Science, July 29(7)* (1973), 363–369.

[38] Fishman, G. Achieving specific accuracy in simulation output analysis. *Communications of the ACM 20*, 5 (May 1977), 310–315.

[39] Fishman, G. Grouping observations in digital simulation. *Management Science 24, 5* (1978), 510–521.

[40] Fishman, G. *Principles of Discrete Event Simulation.* Wiley:NY:UK, 1978.

[41] Fishman, G., and Moore, L. An exhaustive analysis of multiplicative congruential random number generators with modulus $2^{31} - 1$. *SIAM J. Sci. Stat. Comput. 7* (1986), 24–45.

[42] Fitzek, F. Qos support in wireless networks using simultaneous mac packet transmission (SMPT). In *Proc. of ASTC 1999 Featuring Applied Telecommunication Symposium (ATS)* (San Diego, USA, Apr. 1999), pp. 185–190.

[43] Fitzek, F., Mota, E., Ewers, E., Wolisz, A., and Pawlikowski, K. An efficient approach for speeding up simulation of wireless networks. In *Proc. Communication Networks and Distributed Systems Modelling and Simulation Conference CNDS'2001* (Phoenix , Arizona, USA, Jan. 2001), pp. 375–381.

[44] Fox, B., Goldsman, D., and Swain, L. Spaced batch means. *Operations Research Letters 10* (1991), 255–263.

[45] Frost, V., LaRue, W., and Shanmugan, K. Efficient techniques for the simulation of computer communication networks. *IEEE Journal on Selected Areas in Communications SAC-6, 1* (1988), 146–157.

[46] Fujimoto, R. Parallel discrete event simulation. *Communications of the ACM 33*, 10 (Oct. 1990), 30–53.

[47] Fujimoto, R. *Parallel and Distibuted Simulation Systems*. John Wiley, 2000.

[48] Gafarian, A.V. Ancker, C., and Morisaku, T. Evaluation of commonly used rules for detecting steady-state in computer simulation. *Naval Research Logistic Quarterly 78* (1978), 511–529.

[49] Geisler, M. The sizes of simulation samples required to compute certain inventory characteristics with stated precision and confidence. *Management Science 10, 2* (1964), 261–286.

[50] Geyer, C., and Thompson, E. Constrained monte carlo maximum likelihood for dependent data. *J. R. Statit. Soc. B 54*, 3 (1992), 657–699.

[51] Glynn, P. Coverage error for confidence intervals arising in simulation output analysis. In *Winter Simulation Conference* (New York, Dec. 1982), ACM Association for Computing Machinery, pp. 369–375.

[52] Glynn, P. Conditions for the aplicability of the regenerative method. Tech. rep., Stanford University, May 1993.

[53] Glynn, P. Regenerative steady state simulation. Tech. rep., Stanford University, May 1993.

[54] Glynn, P. Some topics in regenerative steady-state simulation. Tech. rep., Stanford University, May 1993.

[55] Glynn, P. The covariance function of a regenerative process. Tech. rep., Stanford University, Apr. 1994.

[56] Glynn, P., and Heidelberger, P. Experiments with initial transient deletion for parallel replicated steady-state simulations. *Management Science 38*, 3 (Mar. 1992), 400–418.

[57] Glynn, P., and Iglehart, D. Simulation output analysis using standardized time series. *Mathematics of Operations Research 15, 1* (1990), 1–16.

[58] Glynn, P., and Whitt, W. Estimating the asymptotic variance with batch means. *Operations Research Letters 10* (1991), 431–435.

[59] Glynn, P., and Whitt, W. The asymptotic validity of sequential stopping rules for stochastic simulations. *The Annals of Applied Probability 2(1)* (1992), 180–198.

[60] Goldsman, D. Simulation output analysis. In *Winter Simulation Conference* (Dec. 1992), IEEE Computer Society Press, pp. 97–103.

[61] Goldsman, D., and Schmeiser, B. Computational efficiency of batching methods. In *Winter Simulation Conference* (Dec. 1997), IEEE Computer Society Press, pp. 202–207.

[62] Goldsman, D., and Schruben, L. Asymptotic properties of some confidence interval estimators for simulation output. *Management Science, Nr. 10, October 30* (1984), 1217–1225.

[63] Gross, D., and Harris, C. M. *Fundamentals of Queueing Theory.* John Wiley & Sons, New York, 1974.

[64] Haynes, L. S., Lau, R. L., Siewiorek, D. P., and Mizel, D. W. A survey of highly parallel computing. *Computer 15*, 1 (Jan. 1982), 9–24.

[65] Heidelberger, P. Statistical analysis of parallel simulations. *Proc. 1986 Winter Simulation Conf. 8-10 dec. 1986* (1986), 290–295.

[66] Heidelberger, P. Discrete event simulations and parallel processing: Statistical properties. *SIAM Journal on Scientific and Statistical Computing 9*, 6 (Nov. 1988), 1114–1132.

[67] Heidelberger, P., and Welch, P. A spectral method for confidence interval generation and run length control in simulations. *Communications of the ACM 24* (1981), 233–245.

[68] Higginbottom, G. *Performance Evaluation of Communication Networks.* Artech House, 1998.

[69] Izydorczyk, J., Mierzwa, J., and Wolisz, A. A control variable method for increasing the efficiency of multiple access protocop simulation. *In Proc. IFIP WG 7.3/TC 6 Second International Symp. Performance of Comput. Commun. Syst.* (1984), 427 – 439.

[70] Kang, K. *Confidence interval estimation via batch means and time series modeling.* PhD thesis, Purdue University, May 1984.

[71] Kelton, W. Random initialization methods in simulation. *IIE Trans. 21* (1989), 355–367.

[72] Kleijnen, J. *Methodology in Systems Modelling and Simulation.* North-Holland, Amsterdam, 1979, ch. The Role of Statistical Methodology in Simulation.

[73] Kleijnen, J. *Statistical Techniques in Simulation.* Marcel Dekker, New York, 1974.

[74] Kleijnen, J. *Statistical Tools for Simulation Practitioners*. Marcel Dekker, New York, 1987.

[75] Kleijnen, J., and Groenendaal, W. *Simulation : A Statistical Perspective*. John Wiley & Sons, 1994.

[76] Kleijnen, J., Ven, V. d., and Saunder, B. Testing independence of simulation subruns : A note on the power of the von neuman test. *European Journal of Operations Research 9* (1982), 92–93.

[77] Kleinrock, L. *Queueing Systems*, vol. I : Theory. John Wiley & Sons, 1975.

[78] Knuth, D. *The Art of Computer Programming, Vol. 2*. Addison-Wesley, 1981.

[79] Kulperger, R., and Lockart, R. Tests of independence in time series. *Journal of Time Series Analysis 19*, 2 (1998), 165–185.

[80] Lavenberg, S., and Sauer, C. Sequential stopping rules for the regenerative method of simulation. *IBM journal of research and development* (1977), 545–558.

[81] Law, A., and Carson, J. S. A sequential procedure for determining the length of a steady-state simulation. *Operations Research 27, 5* (1979), 1011–1025.

[82] Law, A., Kelton, W., and Koenig, L. Relative width sequential confidence intervals for the mean. *Commun. Statist.-Simula. Computa B10*, 1 (1981), 29–39.

[83] Law, A., and KeltonW.D. *Simulation Modeling and Analysis*, 2nd ed. M-H series in industrial engineering and management science. McGraw-Hill, Inc., 1991.

[84] L'Ecuyer, P., and Champoux, Y. Importance sampling for large atm-type queueing networks. pp. 309–316.

[85] Lee, J., McNickle, D., and Pawlikowski, K. A survey of confidence interval formulae for coverage analysis. Tr–COSC 04/98, Department of Computer Scice, University of Canterbury, Christchurch, New Zealand, 1998.

[86] Lee, J.-S. R. *On Automated Sequential Steady-State Simulation*. PhD thesis, University of Canterbury, 2000.

[87] Lee, J.-S. R., Pawlikowski, K., and McNickle, D. Sequential steady-state simulation: Rules of thumb for improving accuracy of the final results. *Proceedings 13th European Simulation Multiconference* (June 1999).

[88] Lee, Y., and Oh, H.-S. Detecting truncation point in steady-state simulation using chaos theory. In *Winter Simulation Conference* (New York, Dec. 2001), ACM Association for Computing Machinery, pp. 353–360.

[89] Meketon, M., and Schmeiser, B. Overlapping batch means: something for nothing ? In *Winter Simulation Conference* (Dec. 1984), IEEE Computer Society Press, pp. 227–230.

[90] Metropolis, M., and Ulam, S. The monte carlo method. *Journal American Statistical Association 44* (1949), 335–341.

[91] Miller, R. The jackknife: A review. *Biometrika 61* (1974), 1–15.

[92] Misra, J. Distributed discrete event simulation. *ACM Computing Surveys 18* (1986), 39–65.

[93] Mota, E., Fitzek, F., Wolisz, A., and Pawlikowski, K. Increasing the accuracy of network simulation experiments. *Proceedings 2000 Advanced Simulation Technologies Conference, Washington* (Apr. 2000), 16–20.

[94] Mota, E., Wolisz, A., and Pawlikowski, K. Sequential batch means techniques for mean value analysis in distributed simulation. *Proceedings 13th European Simulation Multiconference, Warsaw* (June 1999), 129–134.

[95] Mota, E., Wolisz, A., and Pawlikowski, K. A perspective of batching methods in a simulation environment of multiple replications in parallel. *Proceedings of the 20001 Winter Simulation Conference, Washington* (Dec. 2000), 62–66.

[96] Nadas, A. An extension of a theorem of Chow and Robbins on sequential confidence intervals for the mean. *Annals of Mathematical Statistics 40* (1969), 667–671.

[97] Neumann, J. v. Distribution of the ratio of the mean square successive difference to the variance. *Ann. Math. Statist. 12* (1941), 367–395.

[98] Ockerman, D. H., and Goldsman, D. The impact of transients on simulation variance estimators. In *Winter Simulation Conference* (1997), pp. 234–239.

[99] Pawlikowski, K. Steady-state simulation of queueing processes: a survey of problems and solutions. *ACM Computing Surveys 22, 2* (1990), 83–121.

[100] Pawlikowski, K. Simulation studies of telecommunication networks and their credibility. In *European Simulation Multionference* (Warsaw, June 1999), Society for Simulation Computer, pp. 349–355.

[101] Pawlikowski, K., and Asgarkhani, M. Sequential procedures in simulation studies of satellite protocols. *Teletraffic Science for New Cos-Effective Systems and Services Vol. 12, Studies in Telecommunication Elsevier Science Publishers* (1989), 1270–1276.

[102] Pawlikowski, K., and McNickle, D. Speedingup quantitative stochastic simulation. Submitted for publication.

[103] Pawlikowski, K., and McNickle, D. Speedingup quantitative stochastic simulation. In *Submitted for publication.*

[104] Pawlikowski, K., McNickle, D., and Ewing, G. Coverage of confidence intervals in steady-state simulation. *Journal Simulation Practice and Theory* (1998), 255–267.

[105] Pawlikowski, K., and Yau, V. Independent replications versus spectral analysis of output data in steady-state simulation of high speed data networks. In *Australian Teletraffic Research Seminar* (1991), pp. 322–330.

[106] Pawlikowski, K., Yau, V., and McNickle, D. Distributed stochastic discrete-event simulation in parallel time streams. In *Winter Simulation Conference* (New York, Dec. 1994), ACM Association for Computing Machinery, pp. 723–730.

[107] PTOLEMY. Regents of the University of California.

[108] Raatikainen, K. Run length control using parallel spectral method. In *Winter Simulation Conference* (New York, Dec. 1992), ACM Association for Computing Machinery, pp. 594–602.

[109] Rayner, J., and Best, D. Smooth tests of goodness of fit, 1989.

[110] Robertazzi, T. *Communication Networks and Systems : Queueing Theory and Performance Evaluation*. Springer-Verlag, 1990.

[111] Samorodnitsky, G., and Taqqu, M. *Stable Non-Gaussian Random Processes*. 1994.

[112] Sanches, S. Abc's of output analysis. In *Winter Simulation Conference* (New York, Dec. 1999), ACM Association for Computing Machinery, pp. 24–32.

[113] Sargent, R., Kang, K., and Goldsman, D. An investigation of finite-sample behavior of confidence interval estimators. *Operations Research 40, 5* (1992), 892–913.

[114] Schmeiser, B. Batch size effects in the analysis of simulation output. *Operations Research 30* (1982), 556–568.

[115] Schmeiser, B., and Song, W. Correlation among estimators of the variance of the sample mean. In *Winter Simulation Conference* (Dec. 1987), IEEE Computer Society Press, pp. 309–317.

[116] Schriber, T., and Andrews, R. A conceptual framework for research in the analysis of simulation output. *Communications of the ACM 24*, 4 (Apr. 1981), 218–232.

[117] Schruben, L. A coverage function for interval estimators of simulation response. *Management Science 26, 1* (1980), 18–27.

[118] Schruben, L. Detecting initialization bias in simulation output. *Operations Research 30, 3* (1982), 569–590.

[119] Schruben, L. Confidence interval estimation using standardized time series. *Operations Research 31, 6* (1983), 1090–1108.

[120] Schruben, L. Sequential simulation run control using standardized time series. In *PCmpScSt18* (1986), pp. 257– 260.

[121] Schruben, L., Singh, H., and Tierney, L. Optimal tests for initialization bias in simulation output. *Operations Research 31, 6* (1983), 1167–1178.

[122] Shapiro, S. S., and Wilk, M. B. An analysis of variance test for normality (complete samples). *Biometrika 19* (1965), 591–611.

[123] Shapiro, S. S., Wilk, M. B., and Chen, M. H. J. A comparative study of various tests for normality. *JASA 63* (1968), 1343–1372.

[124] Stacey, C., Pawlikowski, K., and McNickle, D. Detection and significance of the initial transient in quantitative steady-state simulation. In *Proc. 8th Australian Teletraffic Research Seminar* (Dec. 1993), pp. 203–212.

[125] Steiger, N., and Wilson, J. Improved batching fo confidence interval construction in steady-state simulation. In *Winter Simulation Conference* (New York, Dec. 1999), ACM Association for Computing Machinery, pp. 442–451.

[126] Trivedi, K. *Probability & Statistics with Reliability, Queueing and Computer Science Applications*. Prentice-Hall, 1982.

[127] Voss, P. A. *Estimating Steady-State Mean from Observed Time Series with Initial Transient*. PhD thesis, Rensseler Polytechinic Institute, Troy, New York, May 1997.

[128] Wagner, D., and Lazowska, E. Parallel simulation of queueing networks: Limitations and potentials. In *Proceedings of 1989 ACM SIGMETRICS and PERFORMANCE '89* (May 1989), vol. 17,1, pp. 146–155.

[129] Wang, J. J., and Abrams, M. Massively parallel simulations with application to queueing networks. Technical Report TR-95-02, Virginia Polytechnic Inst. and State University, Feb. 27, 1995.

[130] Welch, P. *Computer Performance Modeling Handbook*. Academic Press Inc., New York, 1983, ch. 6 : Analytical Results for Queueing Models, pp. 267–329.

[131] Welch, P. On the relationship between batch means, overlapping batch means and spectral estimation. In *Winter Simulation Conference* (Piscataway, NJ, Dec. 1987), ACM Association for Computing Machinery, pp. 309–317.

[132] Whitt, W. The efficiency of one long run versus independent replications in steady-state simulation. *Management Science, Nr. 6 37* (1991), 645–666.

[133] Yeh, Y., and Schmeiser, B. Simulation output analysis via dynamic batch means. In *Winter Simulation Conference* (New York, Dec. 2000), ACM Association for Computing Machinery, pp. 637–645.

[134] Yeh, Y., and Schmeiser, B. On the mse robustness of batching estimators. In *Winter Simulation Conference* (New York, Dec. 2001), ACM Association for Computing Machinery, pp. 344–347.