INITIAL TRANSIENT PERIOD DETECTION USING PARALLEL REPLICATIONS

Falko Bause, Mirko Eickhoff
LS Informatik IV
Universität Dortmund
D-44221 Dortmund, Germany
E-Mail: {bause,eickhoff}@ls4.cs.uni-dortmund.de

KEYWORDS

Steady-State Simulation, Output Analysis, Replication/Deletion Approach, Initialisation Bias, Non-Ergodicity

ABSTRACT

This paper describes an extension of the replication/deletion approach for removal of the initialisation bias in the output data of discrete-event simulations. The proposed extension adaptively determines the cutoff index and also respects a user given ratio between the lengths of transient and steady-state period. We show the benefits of the extension inspecting the well-known M/M/1 queue. Using an example from the logistic area, we also point out that the proposed replication/deletion approach is additionally capable of detecting non-ergodic systems, which is a very difficult task for the usual single long-run simulation.

The presented examples show that the extended replication/deletion approach leads to about the same qualitative steady-state estimates than single long-run simulations, but needs less time in practice.

1 INTRODUCTION

When analysing discrete-event dynamic systems (DEDS) one is often interested in their steady-state behaviour. A very frequently used analysis technique for DEDS is simulation which aims at mimicking the system behaviour on the basis of some state-based model descriptions. Analysis is done by evaluating the observed behaviour.

A significant problem of steady-state analysis concerns the removal of the initialisation bias [Alexopoulos and Seila, 2001], which denotes the influence of the initial state on the simulation results. A common approach, trying to avoid this initialisation bias, is the removal of the first l observations from the simulation results, so that analysis is hopefully based only on observations in steady-state. Surely the main task is to determine the cutoff index l, keeping accuracy of the results and computational effort of the simulation in balance. Many suggestions have been published all being based on the output analysis of a single simulation run (cf. [Pawlikowski, 1990, Schruben, 1982, Schruben, 1983]).

Other approaches for determination of the cutoff index use several, e.g k, independent replications. Welch [Welch, 1983], for instance, computes the averages across several independent replications, in this way constructing a new time series for visual inspection and the user subjectively decides on the cutoff index l. Afterwards the first l observations from the overall n observations of each replication are deleted. This method is commonly known as the replication/deletion method. The number of observations for each replication is normally assumed to be constant and a common advise is to base the decision on the cutoff index and calculation of results on different replications [Law and Kelton, 2000]. In case the resultant $k*(n-l)$ observations do not lead to (statistically) satisfactory results usually further replications are started and one has to keep in mind that for validity of results $l/\ln k \rightarrow \infty$ as $k \rightarrow \infty$ [Fishman, 2001].

Nowadays increasing availability of distributed computing power makes execution of a large number of independent replications more and more feasible. E.g., in local area networks most of the connected computers are usually temporarily idle or less utilised and thus the time exposure for several replications is often not significantly higher compared to a single simulation run on one computing facility.

Proceedings 14th European Simulation Symposium
A. Verbraeck, W. Krug, eds. (c) SCS Europe BVBA, 2002

This offers the possibility to start about a hundred of replications, so that only l and n need to be determined. Law/Kelton [Law and Kelton, 2000] suggest to choose n much larger than l. This can be forced, if there is a specified ratio $1 : r$ between these parameters. With that the only problem is the appropriate choice of l.

In this paper we propose an extension of the replication/deletion method which adaptively determines the cutoff index l on the fly taking a user-specified ratio $1 : r$ between transient and steady-state period into account. The results presented in this paper are excerpts from [Eickhoff, 2002].

The outline of the paper is as follows. Sect. 2 introduces basic notions. Our variant of the *replication/deletion approach* is presented in Sect. 3 and Sect. 4 illustrates its benefits by presenting simulation results of an M/M/1 model, which is theoretically well examined and generally used as a reference example for simulation output analysis approaches [Gafarian et al., 1978, Lee et al., 2000].

2 STEADY-STATE SIMULATION

In steady-state simulation the model is initialised with a system state I. The simulation output can be viewed as a realisation of a stochastic process $(X_i : i \in \mathbb{N})$ and $F_i(x|I) := \Pr[X_i \leq x|I]$ is the state distribution at model time index i. Assuming an ergodic system $F_i(x|I) \to F(x)$ for $i \to \infty$.

The simulation output $x_1, \ldots, x_n$ is a realisation of the random variables $X_1, \ldots, X_n$ and used to estimate $F(x)$ or derived steady-state measures. The first random variables $X_1, \ldots$ are biased due to the influence of the initial system state I. Therefore the removal of this initialisation bias is necessary in order to get a valid estimation of $F(x)$.

Many different rules and tests concerning the removal of the initialisation bias can be found in the literature (cf. [Law and Kelton, 2000, Pawlikowski, 1990]). Gafarian/Anker/Morisaku [Gafarian et al., 1978] and Lee/McNickle/Pawlikowski [Lee et al., 2000], e.g., inspect the quality of some of this rules and tests.

The *replication/deletion approach* is based on the availability of several different replications. In practice different replications are generated by starting simulations with different seeds. An advantage of several random number generators is that appropriate seed values can be calculated beforehand, so that all replications can be executed independently and in parallel. This offers the big advantage that the execution time of all k replication runs scales with the number p of processors/computers involved, provided $k > p$. If the random number generator is a linear congruential generator with no additive constant, i.e. realisation z_i is calculated via $z_i = (a * z_{i-1}) \bmod m$, then the ith realisation can be simply calculated from the initially chosen seed value z_0 by $z_i = (a^i * z_0) \bmod m$. Surely, the implementation has to take possible overflows into account. In order to exploit the full period P of the random number generator for the k independent replications, one can choose $z_{\lceil (j-1)*\frac{P}{k} \rceil}$ as the seed value for the jth replication. Several random number generators have been examined and there are quite well-designed ones with huge period length and numerous sub-streams serving the purpose of generating independent replications excellent [L'Ecyer et al., 2002].

Let $(x_{i1}, \ldots, x_{in})$ denote the output of the ith replication $(i = 1, \ldots, k)$. If $l, l < n,$ is the cutoff index indicating the end of the transient period the sample means in steady-state of the ith replication are given by $y_i(l,n) = \frac{1}{(n-l)} \sum_{j=l+1}^{n} x_{ij}$ from which a point estimate $y(k,l,n) = \frac{1}{k} \sum_{j=1}^{k} y_i(l,n)$ and a confidence interval can be computed (cf. [Alexopoulos and Seila, 2001]).

In the following we show how the cutoff index l can be determined adaptively.

3 ADAPTIVE REPLICATION/DELETION APPROACH

This section describes an adaptive replication/deletion approach, where the cutoff index l is determined on the fly.

Before we describe the algorithm in detail, the following part concerning a possible graphical visualisation of the replication data x_{ij} is intended to support the main idea of the algorithm.

3.1 Graphical Representation

In most cases it is useful to have a visual impression of the simulation data. In the context of many replications, the graph depicting the arithmetic mean of all replications is already an abstraction of the data. In some cases it is better to have a visualisation of the distribution derived from multiple replications. The data for this visualisation can be derived by splitting the model time into *X-intervals* (abscissa). Measures like, population, entry-rate, exit-rate etc. can then be computed for every X-interval of each replication.

Moreover the range of the observed measures can be divided into equally spaced *Y-intervals* (ordinate). The output data can be assigned to these Y-intervals. With this assignment a frequency-histogram for each X-interval can be computed. So the original data is transformed to an area of several *X-Y-boxes*, which contain the frequency of the data values.

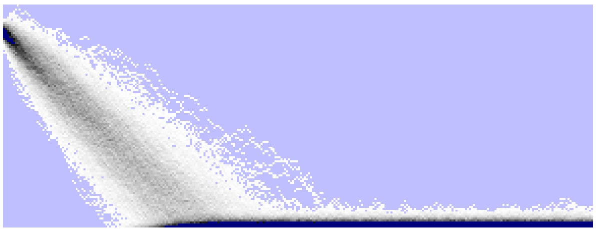

Figure 1: Population at a high initialised M/M/1-server (1000 replications). (abscissa: model time; ordinate: population state; grey-intensity: frequency)

To give a visual impression of this area and the dynamic behaviour, we use a pixel for every X-Y-box. The model time is on the abscissa and the range of the measure is on the ordinate. The value of the frequency is represented by a grey-intensity. In general *white* means the frequency value 0 and *black* means the frequency value 1. Lower and upper bounds give the possibility to emphasise a specific frequency range. So the available 256 gray-intensity values can be used to represent all frequency values in between the bounds. All frequency values smaller than the lower bound are displayed in a light blue, whereas all frequency values beyond the upper bound are shown in dark blue. So Fig. 1 is a 2-dimensional histogram, where grey-intensities are chosen to represent the different levels, instead of using a 3-dimensional graph.

3.2 Data Transformation

The simulation is event-driven and the events occur at different model times in each replication. So the event-streams of the replications cannot be directly compared with each other and it is thus necessary to transform the event-stream $e_1, \ldots, e_m$ to output values $x_1, \ldots x_n$.

As mentioned before, the model time is split into X-intervals TI_j with $j = 1, \ldots, n$ of size t_I. The measures x_{ij} can be computed for all these X-intervals and for each replication separately.

E.g., when computing the exit-rate of X-interval i, one counts all the exit-events of this interval divided by its size, i.e. $x_{ij} =$ (#exits in TI_j at replication $i)/t_I$. Similarly, the entry-rate, the population, etc. can be computed.

3.3 Basic Idea of the Algorithm

In the usual replication/deletion approach the cutoff index l (truncation point) must be chosen by the analyst. The graphical representation of the simulation data can give a hint for a proper choice. Surely, a statistical analysis is more desirable.

Let k denote the number of replications, n the amount of observed X-intervals and let x_{ij} $(i = 1, \ldots, k; j = 1, \ldots, n)$ be the transformed output of the replications. Then $x_{1j} \ldots, x_{kj}$ is the random sample of the jth X-interval: RS_j.

In steady-state there is no change in the probability distribution and the graphical representation will show no change in the run of all curves. Provided l is the proper cutoff index, all RS_j with $j = l, \ldots, n$ are realisations from the same probability distribution.

The assumption of a steady-state might be unreliable, if the cutoff index l is near to the total amount of observed data n. Law/Kelton [Law and Kelton, 2000], e.g., suggest to choose n much larger than l, but in many analysis methods l is chosen independently of n. Our algorithm pays attention to a user-specified ratio between l and n, so that the chances are better to actually select the steady-state period.

3.4 Algorithm

The cutoff index l can be estimated with statistical methods. In this section an algorithm is proposed, that works on the fly on the generated data.

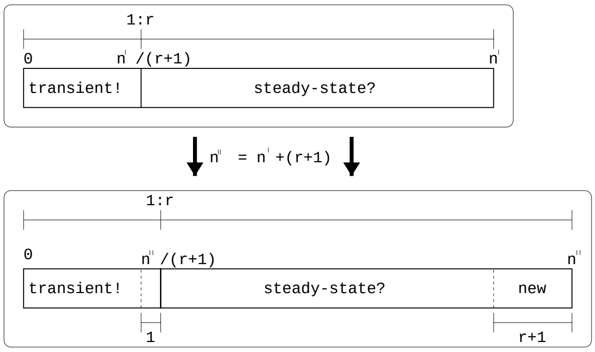

Figure 2: Data collection according to ratio $1 : r$

Let TS denote the test-sample RS_j, when testing whether j is a proper estimate of l. A safety-level p is needed to consider the error of the statistical comparison-method (details later).

1. Choose the ratio $1 : r$, the number of replications k, the size of the X-intervals t_I and a safety-level $0 \leq p \leq 1$.
2. Initialisation: $TS := RS_0$ and $n := 0$
3. Observe $r + 1$ new X-intervals of all replications and compute the $r + 1$ new random samples: $RS_{n+1}, \ldots, RS_{n+r+1}$
4. $n := n + (r + 1)$ — the amount of data is increased
5. $TS := RS_{\frac{n}{r+1}}$ — select a new test sample
6. Compare TS with RS_j for $j = \frac{n}{r+1} + 1, \ldots, n$.
   If more than $(p * 100)$% of the compared random samples RS_j have a different probability distribution than TS: goto 3.
7. Calculate the result values as usual with cutoff value $l := \frac{n}{r+1}$ (see Sect. 2).

In the following we will explain some parts of the algorithm in more detail.

Details to 6.: Many statistical standard tests can be used to compare two random samples (e.g.: Wilcoxon Signed-Rank Test, Mann-Whitney U-Test, Kolmogoroff-Smirnov-Test, χ^2-Test). We used the Kolmogoroff-Smirnov two-sample test (cf. [Hartung et al., 1985]), that checks, whether two independent samples are from the same distribution. The test uses the maximal difference between the cumulative distributions F_1 and F_2 of the two random samples. The null hypothesis is:

$$H_0 : F_1(x) = F_2(x) \quad (1)$$

In our case the Kolmogoroff-Smirnov two-sample test seems to be reasonable, because no assumptions on F_1 and F_2 must be given and the critical points can be determined efficiently. One can choose between different α-levels. For the examples given in this paper we used the α-level 0.05.

Details to 7.: What options do one has, if the estimated parameters of the replication/deletion approach do not lead to a sufficiently small confidence interval? There are two ways to solve this problem:

1. Increase k: As described by Fishman ([Fishman, 2001]), this will have an effect on the cutoff index and so l must be determined again. The less time-consuming alternative is:
2. Increase n: This has no effect on the cutoff index and due to the *law of large numbers* the confidence interval will get smaller in a stable system.

Increasing n can easily be realised by continuing the execution of the replications. From this point of view, the replication/deletion approach is used as a sequential procedure [Law and Kelton, 2000], that stops, if confidence is acceptable.

Parameters: The adaptive replication/deletion approach needs some parameters:

k**:** A good choice for k depends on the chosen statistical method for random sample comparisons. Generally speaking it is surely advisable to make as much replications as the hardware can handle.

r**:** A larger r results in a better estimation of l. A good choice for r surely depends on the model (see Sect. 4.2).

t_I**:** A good choice for t_I also depends on the model. Increasing t_I implies smoother measures.

p**:** This parameter considers the error of the statistical method for random sample comparisons.

4 EXAMPLES

We used the standard M/M/1 model in order to test the accuracy of our version of the replication/deletion approach. The output analysis of this model shows typical problems (see [Gafarian et al., 1978, Lee et al., 2000]). In this section the results of several output analyses are compared with each other and with theoretical results.

4.1 M/M/1

An M/M/1-server is stable, when the utilisation $\rho := \frac{\lambda}{\mu}$ is less than 1. In this case the mean number of jobs in the system (population) in steady-state is determined by $E[N] = \frac{\rho}{1-\rho}$. In the following examples we choose $\lambda := 1$ and $\mu := \frac{1}{0.8}$ resp. $\frac{1}{0.95}$.

Utilisation: 0.8 — Initialisation: 100

First we use a M/M/1-server that is utilised on a "normal" level ($\rho = 0.8$; $E[N] = 4$). The system is initialised with 100 jobs, which results in an obvious transient period.

Fig. 3 and 4 show the results of the adaptive replication/deletion approach with $k = 100$, $t_I = 20$ and $r = 3$.

The graph in Fig. 3 shows the comparisons with TS for different time steps. In the beginning the model time of TS

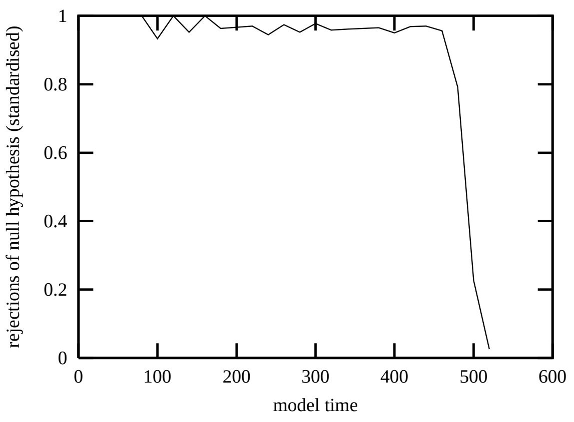

Figure 3: Graph of the comparisons with TS

is 0 and at the end it is 540. The amount of data observed to estimate $l = \frac{540}{t_I} = 27$ is $n = (l-1)*(r+1) = 104$ (model time: $104 * 20 = 2080$). The range on the ordinate is from 0 to 1 and represents the frequency of the rejections of the null hypothesis H_0 from the Kolmogoroff-Smirnov two-sample test standardised by the number of comparisons.

The graph starts on a very high level close to 1 implying, that the null hypothesis H_0 is rejected in nearly all comparisons of TS with RS_j. So there is a transient period till the model time 520. As soon as TS enters the steady-state period (480 to 520) the graph drops down to 0. At model time 540 the steady-state period is reached.

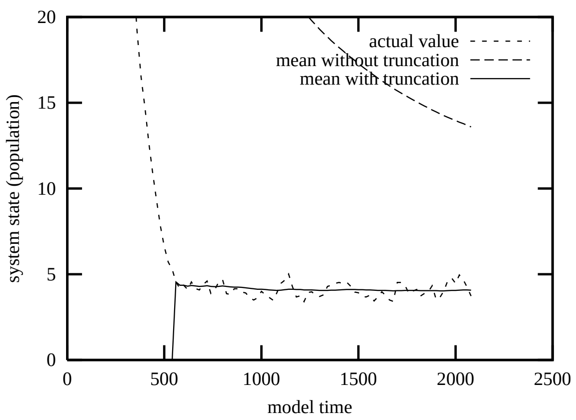

Figure 4: The effect of initial truncation at 540 ($\rho = 0.8$;Init.: 100).

In Fig. 4 the first graph shows the actual state of the observed system measure. The second graph shows the running mean [Pawlikowski, 1990] of the actual state without initial truncation. The significant initialisation bias is obvious. In contrast, the third graph shows the running mean of the actual state with the cutoff index l. No initialisation bias is visible and the graph is nearly a straight line without a slope.

approach	$E[N]$ **est.**	$l * t_I$	$n * t_I$
ada. rep/del	$4.09 \pm 4.7\%$	540	2080
single1	$11.08\pm$ undef		2080
single2	$4.16 \pm 4.5\%$		208000
single3	$4.16 \pm 4.9\%$	5000	136000

Figure 5: Results of 4 different simulations of the M/M/1-server with $\rho = 0.8$ and 100 jobs initially.

Fig. 5 shows a table of 4 different simulations. *ada. rep/del* denotes the simulation runs using the adaptive replication/deletion approach. *single1* and *single2* are output analyses of a single simulation without initial truncation. In *single3* the transient period is determined with *Rule 6* described in [Pawlikowski, 1990] and cut off. The estimated values are given with a confidence level of 95%.

ada. rep/del and *single1* need the same amount of model time, but the estimation from *single1* is useless. *single2* and *single3* give an estimation with the same quality, but they need much more time than *ada. rep/del*. *single3* needs the least amount of data ($k * 2080 > 136000$), but *ada. rep/del* collects the data from parallel replications and is thus faster in practice.

Utilisation: 0.8 — Initialisation: 0

For the next analysis we used the same M/M/1-model, except the initialisation. Now the model is initialised *empty & idle*.

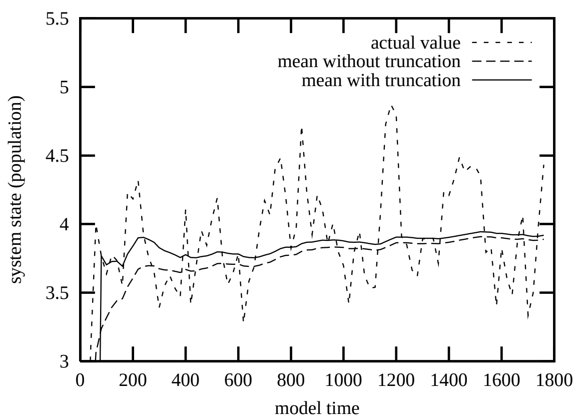

Figure 6: The effect of initial truncation at 60 ($\rho = 0.8$;Init.: 0).

Like Fig. 4, Fig. 6 gives an impression of the difference between an output analysis with and without initial truncation. Because of the new initial state (*empty & idle*) the difference is not as significant as in the former example. Since the chosen initial state is closer to $E[N]$ we have a shorter transient period.

approach	$E[N]$ **est.**	$l * t_I$	$n * t_I$
ada. rep/del	$3.92 \pm 4.5\%$	60	1760
single1	$3.08 \pm 24.3\%$		1760
single2	$4.10 \pm 3.9\%$		176000
single3	$4.15 \pm 5.0\%$	5200	138400

Figure 7: Results of 4 different simulations of the M/M/1-server with $\rho = 0.8$ and 0 jobs initially.

ada. rep/del, *single1*, *single2* and *single3* are the same approaches as described before. For the adaptive replication/deletion approach we selected $k = 100$, $t_I = 20$ and $r = 3$. So for the estimation of $l = \frac{60}{t_I} = 3$ only the data up to model time 160 is needed. The rest of the collected data is used to get a smaller confidence interval.

The results (see Fig. 7) show the same facts: *single3* needs the least amount of data ($k * 1760 > 138400$), but *ada. rep/del* is much faster, because of the parallel replications.

Utilisation: 0.95 — Initialisation: 100

The estimation of performance measures in a high utilised system is well-known to be more difficult, because in most cases the variation in the range of the system state is larger. This section describes the analysis of an M/M/1-model with $\rho = 0.95$ (implying $E[N] = 19$). The model is again initialised with 100 jobs.

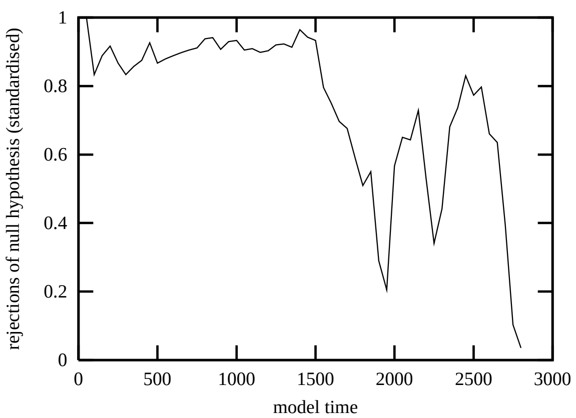

Figure 8: Graph of the comparisons with TS

Fig. 8 is similar to Fig. 3 in most parts, two facts are different: A) The transient period is longer, because the server needs more time to work off the initial tailback. B) The graph is not falling such smooth as before, because of a larger range of variation of the system state.

Again Fig. 9 shows a significant difference between the output analysis with and without initial truncation.

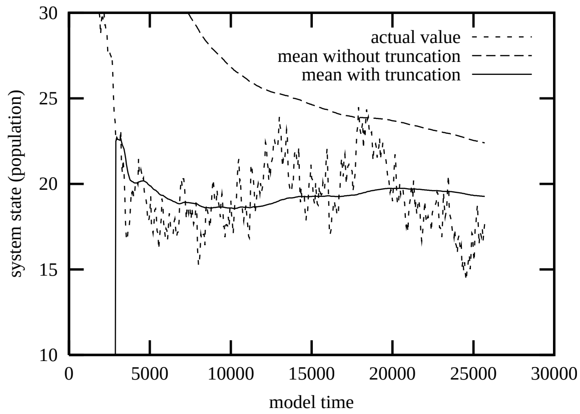

Figure 9: The effect of initial truncation at 2850 ($\rho = 0.95$;Init.: 100).

ada. rep/del, *single1*, *single2* and *single3* are the same approaches as described before. The parameters for the adaptive replication/deletion approach are $k = 100$, $t_I = 50$ and $r = 3$. The results (see Fig. 10) show, that more data is needed for a proper result, but no serious problems result from the high utilisation.

approach	$E[N]$ **est.**	$l * t_I$	$n * t_I$
ada. rep/del	$19.28 \pm 5.0\%$	2850	25700
single1	$30.17 \pm 44.4\%$		25700
single2	$19.17 \pm 3.5\%$		2570000
single3	$19.04 \pm 5.0\%$	3500	965800
single0	$104.47 \pm 1.1\%$		150

Figure 10: Results of 5 different simulations of the M/M/1-server with $\rho = 0.95$ and 100 jobs initially.

The additional simulation *single0* serves as an educational example in this context. It is a single simulation run without initial truncation, that stops when the confidence interval (level 95%) is smaller than 1.5% of the arithmetic mean. This approach will lead to a wrong estimation, because the trend in the first data is not detected and the data until model time 150 is smooth enough to meet the stop-criteria.

4.2 A Non-Ergodic System

In this section we use a model that is described in detail in [Bause and Beilner, 1999]. The main characteristic of this model is its (structurally) non-ergodicity, which is very difficult to detect when simulating. Bause and Beilner have shown that a single-long-run simulation might affirm ergodicity for a very long model time.

In Fig. 11 the possible pitfall is obvious. The confidence interval is very close to the mean. If a stop-criteria for the

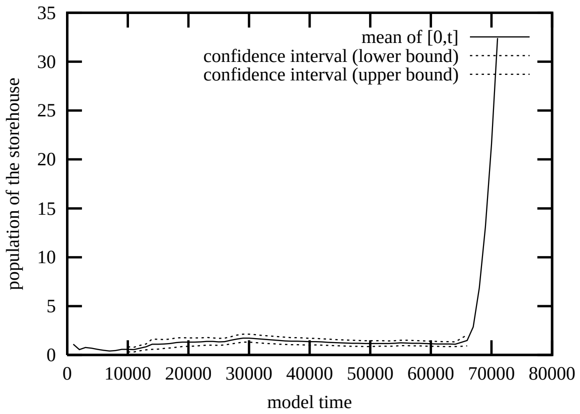

Figure 11: The population at the storehouse computed of one single simulation run.

simulation is based on the confidence interval, the pitfall is nearly unavoidable. A mean will be calculated, although the system is not ergodic.

For this example the adaptive replication/deletion approach is executed with $k = 100$, $t_I = 200$ and $r = 9$. We choose larger values for t_I and r to make sure that observations beyond a possible end of the transient period are observed.

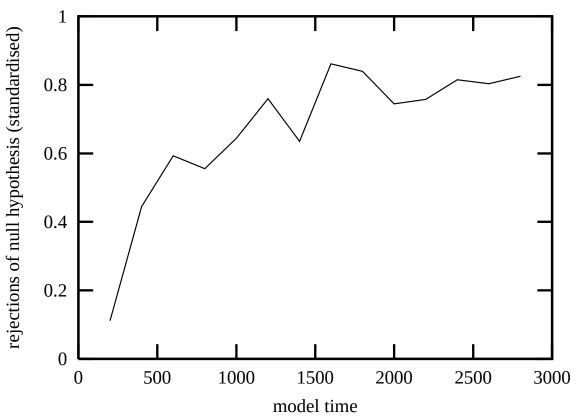

Figure 12: Graph of the comparisons with TS

Fig. 12 shows the comparison of TS with the following RS_j. In general the graph shows a bigger discrepancy, the later the model time at TS is. The latest model time observed is $n = 2800 * (r + 1) = 28000$. So either the model has a very long transient period or is not stable.

This result is much better than the result of an approach with a stopping condition based on a confidence interval of a single run, since the output data might satisfy this condition as shown in [Bause and Beilner, 1999].

5 CONCLUSIONS

In general the output analysis of a single run needs less data, because the transient period is only traversed once. Even though, the output analysis of multiple replications is often much faster, since all replications can be executed in parallel and directly scale with the number of processors/computers involved.

Apart from statistical tests, graphical representations of the output data (see Fig. 1) can often help to find a proper cutoff index. In such cases the choice of a high initialisation state might support this decision, which in a way follows the advise of Kelton/Law [Kelton and Law, 1985]: "... the optimal state for initialization tends to be larger than the mean ...".

In our algorithm the analyst is forced to choose a ratio between the lengths of the transient period and the observed part of the steady-state period, thus avoiding that the independent choice of both parameters leads to poor results.

Most detection methods/rules (e.g. the graphical procedure of Welch [Welch, 1983]) are based on the convergence of the mean to its steady-state value. But what, if only the mean converges and for example other quantiles do not?! The adaptive replication/deletion approach checks the random sample distribution and is therefore a better test for the ergodicity of the system. As shown, even non-ergodic models can be detected more easily with the adaptive replication/deletion approach.

6 FUTURE WORK

In general, every user-specified parameter of a method for output analysis is a possible source of error, especially for an unexperienced user. So one of our aims is to reduce the amount of such parameters. An idea that refers to this problem is to choose the ratio $1 : r$ adaptively as well as the cutoff index. The ratio can be varied depending on the smoothness of the standardised number of rejections of the null hypothesis.

An M/M/1-server is a well understood and therefore ideal reference model. But we are looking forward to see, how the adaptive replication/deletion approach behaves on real-world systems. We want to collect a useful selection of models for comparison with other replication/deletion methods, like the graphical procedure of Welch [Welch, 1983].

REFERENCES

[Alexopoulos and Seila, 2001] Alexopoulos, C. and Seila, A. F. (2001). Output data analysis for simulations. *Proceedings of the 2001 Winter Simulation Conference*, pages 115–122.

[Bause and Beilner, 1999] Bause, F. and Beilner, H. (1999). Intrinsic problems in simulation of logistic networks. *11th European Simulation Symposium and Exhibition (ESS99)*, pages 193–198.

[Eickhoff, 2002] Eickhoff, M. (2002). Statistische Auswertung und Erkennung der stationaeren Phase in der Simulation zustands-diskreter Systeme. Diploma Thesis, Universitaet Dortmund - Fachbereich Informatik - Lehrstuhl IV.

[Fishman, 2001] Fishman, G. S. (2001). *Discrete-Event Simulation*. Springer.

[Gafarian et al., 1978] Gafarian, A. V., Ancker, C. J., and Morisaku, T. (1978). Evaluation of commonly used rules for detecting steady state in computer simulation. *Naval Research Logistics Quarterly*, pages 511–529.

[Hartung et al., 1985] Hartung, J., Elpelt, B., and Kloesener, K.-H. (1985). *Statistik*. R. Oldenbourg Verlag.

[Kelton and Law, 1985] Kelton, W. D. and Law, A. M. (1985). The transient behavior of the m/m/s queue, with implications for steady-state simulation. *Operations Research*, 33:378–396.

[Law and Kelton, 2000] Law, A. M. and Kelton, W. D. (2000). *Simulation Modeling and Analysis*. McGraw-Hill Higher Education.

[L'Ecyer et al., 2002] L'Ecyer, P., Simard, R., Chen, E. J., and Kelton, W. D. (2002). An object-oriented random-number package with many long streams and substreams. *To appear in Operations Research*, 50(6).

[Lee et al., 2000] Lee, J.-S. R., McNickle, D., and Pawlikowski, K. (2000). Initial transient period detection for steady-state quantile estimation. *Proceedings of the Summer Computer Simulation Conference*, pages 169–174.

[Pawlikowski, 1990] Pawlikowski, K. (1990). Steady-state simulation of queueing processes: a survey of problems and solutions. *ACM Computing Surveys*, 22:123–170.

[Schruben, 1982] Schruben, L. W. (1982). Detecting initialization bias in simulation output. *Operations Research*, 30(3):569–590.

[Schruben, 1983] Schruben, L. W. (1983). Confidence interval estimation using standardized time series. *Operations Research*, 31:1090–1108.

[Welch, 1983] Welch, P. D. (1983). The statistical analysis of simulation results. *In The Computer Performance Modeling Handbook, ed. S. Lavenberg, Academic Press*, pages 268–328.

ACKNOWLEDGEMENTS

This research was supported by the Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center "Modelling of large logistics networks"(559).

BIOGRAPHY

DR. FALKO BAUSE has taught and done research work in the area of system engineering with emphasis on Stochastic Petri Nets. He initially defined the Queueing Petri Net formalism, which combines Queueing Networks with Stochastic Petri Nets. He also coauthored a book with the title "Stochastic Petri Nets – An Introduction to the Theory". He currently leads a project concerning the modelling and analysis of logistic nets supported by the Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center "Modelling of large logistics networks" (559).

MIRKO EICKHOFF holds a Diploma degree (Dipl.-Inform.) in Computer Science from the University of Dortmund. Since the beginning of 2002 he has taught in operating and distributed systems. He has done research work in the area of system modelling and output analysis with emphasis on simulation techniques using multiple replications. Currently he is involved in a research project concerning the modelling and analysis of logistic nets supported by the Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center "Modelling of large logistics networks" (559). In this context his interests concern statistical methods for output analysis.