Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 173 (2006) 252–267
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Decision Support

Analysis of the behavior of the transient period in non-terminating simulations

Burhaneddin Sandıkçı [a], İhsan Sabuncuoğlu [b,*]

[a] Department of Operations Research, University of North Carolina, Chapel Hill, NC 27599-3180, USA
[b] Department of Industrial Engineering, Faculty of Engineering, Bilkent University, Bilkent, Ankara 06533, Turkey

Received 6 June 2004; accepted 26 November 2004
Available online 16 February 2005

Abstract

Computer simulation is a widely used tool for analyzing many industrial and service systems. However, a major disadvantage of simulation is that the results are only estimates of the performance measures of interest, hence they need careful statistical analyses. Simulation studies are often classified as either terminating or non-terminating. One of the major problems in non-terminating simulations is the problem of initial transient. Many techniques have been proposed in the literature to deal with this problem. There are currently a number of studies to improve the efficiency and effectiveness of these techniques. However, no research has been reported yet that analyzes the behavior of the transient period. In this paper, we investigate the factors affecting the length of the transient period for non-terminating simulations, particularly for serial production lines and job-shop production systems. Factors such as the variability of processing times, system size, existence of bottleneck, reliability of system, system load level, and buffer capacity are investigated. Recommendations for the use of a new technique are given. A comprehensive bibliography is also provided.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Nonterminating simulations; Behavior of transient period

1. Introduction

The idea of modeling is one of the most important ways of studying, understanding, and improving the behavior of either existing or to be built systems. Among several modeling approaches available today, *simulation modeling* receives increased attention from both practitioners and

* Corresponding author. Tel.: +90 312 266 4477; fax: +90 312 266 4126.
E-mail addresses: sandikci@email.unc.edu (B. Sandıkçı), sabun@bilkent.edu.tr (İ. Sabuncuoğlu).

0377-2217/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2004.11.021

academics, as the complexity of the systems increases (Harpell et al., 1989). Many simulation models built today are stochastic simulation models. Two problems with stochastic simulation output are often discussed in the literature: non-stationarity and autocorrelation. *Non-stationarity* means that the distributions of the successive observations in the output sequence change over time. *Autocorrelation* means that the observations in the time sequence are correlated with each other. So the classical statistical assumption of independently and identically distributed (iid) outputs/observations is violated.

Simulation experiments are classified as either terminating or non-terminating as far as the goal of the simulation is concerned (Law and Kelton, 2000; Fishman, 2001). The above stated problems do not exist in terminating simulations since the underlying system explicitly determines the starting and stopping conditions for the simulation model. Hence, the method of independent replications is commonly used for these simulations, resulting in iid observations. A *non-terminating simulation*, on the other hand, aims to estimate the steady-state parameter(s) of a system. However, the practical simulation, which starts and ends at a user-defined state, may cause inaccurate results if the initial conditions are not chosen from the steady-state. This is called the *initial transient*, *initialization bias*, or *the start-up problem* in the simulation literature. Several techniques have been proposed to remedy this problem (see, for example, Kelton, 1989; Kelton and Law, 1983; Schruben, 1982; Schruben et al., 1983; Goldsman et al., 1994; Vassilacopoulus, 1989; Welch, 1982; White, 1997).

The primary motivation for this study comes from the negligence of initial transient problem in practice. The effect of this negligence is severe, especially when using the method of independent replications, since the initialization bias is not affected by the number of replications but by the length of each run or by the amount of truncation per run. The lack of objective procedures to deal with the initial transient problem that are guaranteed to work well in every situation is another motivation for this study. The common practice is to truncate some initial portion of the output sequence; however, this is done in a rather informal way. Furthermore, in system comparisons and optimization studies, the truncation point is usually chosen by observing only one particular scenario, which could be a poor sample in terms of the transient period; and the same amount of data is truncated from all other simulated scenarios.

Almost all of the studies in the literature either develop methods or compare the effectiveness of proposed techniques via their application to analytically tractable models. We have not encountered any study that explicitly investigates how the initial transient period behaves with respect to different system parameters. If some guidelines could be given, then the problems discussed above would be alleviated—if not completely eliminated. In this paper, we are primarily interested in the *behavior* of the initial transient with respect to changes in the system parameters.

To be more specific, we focus on manufacturing systems; particularly serial production lines and job-shop production systems. The reason for choosing these systems is that they are the building blocks of most manufacturing systems, and one can observe the simplest form of interactions among system components, which then can be generalized to larger systems. Additionally, these systems are still widely used in practical manufacturing. Our results are meant to provide a framework for simulation practitioners to validate their model findings regarding the transient period. Moreover, we test a relatively new truncation technique (*MSER*) to assess its theoretical limitations and to give some guidelines for its successful implementation.

The rest of the paper is organized as follows. A comprehensive literature review is given in Section 2. This is followed by the methodology of this study in Section 3. Section 4 presents experimental factors and conditions. Simulation results are discussed in Section 5. Concluding remarks are given in Section 6.

2. Literature review

The problem of initial transient has been investigated by many researchers. The literature can be

Table 1
Summary of the literature on the initial transient problem

Type of study	Studies conducted
General	Gafarian et al. (1978), Wilson and Pritsker (1978a,b), Chance (1993), Fishman (1972), Kleijnen (1984), Law (1984), Nelson (1990, 1992), Cash et al. (1992), Ma and Kochhar (1993)
Intelligent initialization	
Deterministic initialization	Madansky (1976), Kelton and Law (1985), Kelton (1985), Murray and Kelton (1988a)
Stochastic initialization	Kelton (1989), Murray (1988), Murray and Kelton (1988b)
Antithetic initial conditions	Deligönül (1987)
Truncation heuristics	
Graphical techniques	Welch (1982)
Repetitive hypothesis testing	Schruben (1981, 1982), Schruben et al. (1983), Goldsman et al. (1994), Vassilacopoulus (1989)
Analytical techniques	Kelton and Law (1983), Asmussen et al. (1992), Gallagher et al. (1996), White (1997), Spratt (1998), White et al. (2000)

divided into three broad categories; (1) general studies, (2) intelligent initialization methods, and (3) truncation heuristics. Table 1 summarizes the literature on initial transient problem, which we discuss now in the order presented in the table.

2.1. General literature

Gafarian et al. (1978) and Wilson and Pritsker (1978a) review various truncation heuristics, and find that the methods available at that time are rather unsatisfactory. Wilson and Pritsker (1978b) state that choosing an initial state near the mode (rather than the mean) of the steady-state distribution produces favorable results. Another survey is provided by Chance (1993). Fishman (1972) uses a first-order autoregressive scheme to demonstrate that initial data truncation reduces bias, but increases variance. Some authors suggest that—for special systems—retaining the whole sequence would minimize the mean-squared-error (MSE) (Kleijnen, 1984). Indeed, Law (1984) proved that—for simple queuing systems—MSE is minimized by using the whole series. Nelson (1992) suggests using fewer replications and longer runs per replication in the presence of initialization bias and a tight budget.

Cash et al. (1992) assess the tests for initial bias detection provided by Goldsman et al. (1994) on analytically tractable models. They report that these tests are powerful when the bias is severe at the beginning of the sequence, and dies out quickly. However, if the bias decays slowly, it becomes harder for the tests to detect the bias. Ma and Kochhar (1993) compare the test procedures of Schruben (1982) and Vassilacopoulus (1989), using sequences with known transient distributions. Their results indicate that both tests are powerful, but they recommend Vassilacopoulus's test due to its ease of implementation. We refer to Nelson (1990) for variance reduction techniques (which is a broad area in itself) in the presence of initialization bias.

2.2. Intelligent initialization

Intelligent initialization simply uses the idea of starting a simulation in a state that is representative of the system's steady-state. This approach can be implemented in two ways. The first is called *deterministic (fixed) initialization*, where the initial conditions are chosen as constant values, such as the mean or the mode of the steady-state distribution. A second way, called *stochastic (random) initialization*, tries to estimate the steady-state probability distribution of the process, possibly from pilot runs, and then uses this estimated distribution to sample the initial conditions.

Madansky (1976) shows that initializing an $M/M/1$ queue in empty and idle state, which is

the mode of the number-in-system distribution, minimizes the MSE of the point estimate. For $M/M/s$, $M/E_m/1$, $M/E_m/2$, and $E_m/M/2$ queues, Kelton and Law (1985), Kelton (1985), and Murray and Kelton (1988a) find that initializing in a state at least as congested as the steady-state mean (as opposed to the mode) induces shorter transient periods.

Kelton (1989) uses the idea of random initialization and finds that it reduces the severity and duration of the initial transient period, compared with starting in a fixed state. He recommends initializing simulations stochastically when having relatively short runs. However, Murray (1988) emphasizes the difficulties of applying this technique in many practical simulations. Also, Murray and Kelton (1988b) use a first-order autoregressive process to show that random initialization is effective in reducing bias. A similar approach is suggested by Deligönül (1987); however, this approach starts with antithetic conditions rather than random conditions.

2.3. Truncation heuristics

Truncation heuristics may be applied to any simulation output sequence. The idea is to delete some observations from the beginning of the sequence that do not represent the steady-state and use only the remaining observations to estimate the quantities of interest. However, truncation is not an easy task at all. Given a biased sequence due to initialization, deleting some initial data will increase the accuracy of the point estimator; on the other hand, extensive truncation would imply a loss of precision. Therefore, users should carefully consider the tradeoff between accuracy and precision. Nevertheless, these methods are more widely accepted than intelligent initialization techniques, due to their simplicity. Truncation heuristics can further be classified as those that directly suggest a truncation point, and those that recursively apply hypothesis testing to detect initialization bias.

One of the simplest and most widely used techniques for determining a truncation point is a graphical procedure due to Welch (1982)—summarized in Law and Kelton (2000)—which is based on making several independent replications and averaging across replications. Further reduction in the variability of the plot can be achieved by moving averages. When the resulting statistics are plotted, the truncation point is chosen to be the point where the graph flattens out.

Schruben (1982) develops a very general procedure for univariate output based on standardized time-series. This procedure is the basic building block of techniques discussed by Schruben et al. (1983) and Goldsman et al. (1994), which we call repetitive hypothesis testing. Given a set of data, the user recursively deletes some data from the beginning, and checks for initialization bias until the test concludes that no bias is left in the sequence. However, this might be a too time-consuming task. Instead one can delete some data via some other technique, and apply this test to the remaining observations to determine if there is any bias left. The theoretical framework for the multivariate case is also given by Schruben (1981). Furthermore, Vassilacopoulus (1989) also proposes a hypothesis test to select the truncation point, but he uses a different test statistic, which is easier to compute than Schruben's statistic.

Kelton and Law (1983) develop an algorithm for simultaneously choosing the truncation point and the run length. Their algorithm is based on linear regression and worked well for a wide variety of stochastic models. However, a practical drawback of the algorithm is that it requires the analyst to set several parameters. Those authors also suggest to start in an undercongested state rather than in an equally overcongested state.

Asmussen et al. (1992) propose several algorithms. They also prove that there does not exist a universally satisfactory means of detecting stationarity in a stochastic sequence—without some restrictions on the class of simulations to be considered. Gallagher et al. (1996) use a Bayesian technique called Multiple Model Adaptive Estimation (MMAE) with three Kalman filters. They select a truncation point when the MMAE mean estimate is within a small tolerance of the assumed steady-state.

Recently, White (1997) proposed a truncation heuristic named the Marginal Confidence Rule (MCR). With almost no modification, White

et al. (2000) renamed it the Marginal Standard Error Rule (*MSER*). They compare this rule to several other heuristics; their results indicate that a variant of *MSER* (namely, *MSER*-5 due to Spratt, 1998) dominates other rules. Claimed advantages of this new rule are its ease of understanding and implementation, inexpensive computation, efficiency in preserving representative simulation data, and effectiveness in mitigating the initial bias. We use this new rule in the next sections.

3. Model building, data collection, and output data analysis

We program our simulation models in AutoMod version 9.1 (1999). Some of our analyses have been programmed in MATLAB version 5.3 (1995). We selected the *time-in-system* statistic for our analyses. We used five independent replications, each replication having 30,000 observations. This run length was determined based on pilot runs (it is long enough to allow the rarest events to occur at least 30 times in the most extreme case). These observations are then batched into groups of five.

We use two truncation heuristics to determine the length of the transient period: the cumulative averages plot (as a graphical approach) and the *MSER* (as a quantitative method). Instead of cumulative averages plot, one can think of using Welch's technique due to its popularity. However, Fig. 1 shows that these two techniques do not produce significantly different results. Besides, Welch's technique requires the analyst to decide on a window size (w) by trial-and-error, which makes it practically less applicable.

We start the cumulative averages plot by calculating the cumulative average ($\overline{X}_k$):

$$\overline{X}_k = \frac{1}{k}\sum_{i=1}^{k} X_i \quad \text{for } k = 1, 2, \ldots, n,$$

where $\{X_i, i = 1, 2, \ldots, n\}$ is the given sequence. Then, $\overline{X}_k$ for $k = 1, 2, \ldots, n$ is plotted against k; in our case $n = 6000$. A truncation point, d, is selected visually such that the curve seems to become nearly horizontal. In Fig. 1(a), we see that truncating 300 observations would be enough in either case (the outliers issue will be discussed in more detail at the end of this section).

The *MSER* heuristic, on the other hand, determines the truncation point by minimizing the standard error (s.e.)

$$\text{s.e.} = \sqrt{\frac{S^2_{n-d}}{n-d}},$$

where S^2_{n-d} is the sample variance of the remaining sequence (n is still the number of observations in the original sequence). The idea is to delete observations one at a time from the beginning of the sequence, and calculate s.e. for the remaining sequence. Once all s.e.'s are calculated, it is suggested to choose the end of the transient period such that s.e. is minimized. Since we batch the original data into groups of 5, we actually apply the rule called *MSER*-5.

The most important advantages of the *MSER* are that it provides quantitative values for the

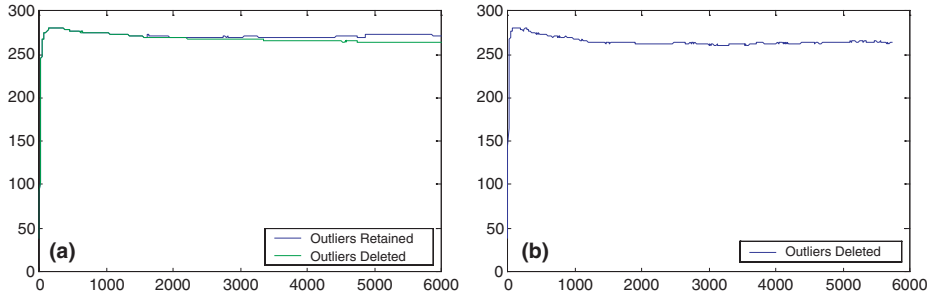

Fig. 1. Cumulative averages plot vs. Welch's graph.

truncation point; it is easy to compute—even for very large samples. However, our experiences reveal two problems. The first one is theoretical, in the sense that the method makes use of the sample variance, S^2_{n-d}, which is calculated from a correlated sequence. It is well-known that autocorrelation might induce significant bias in the variance estimation, which means s.e.'s will also be biased; see Law and Kelton (2000, pp. 530–531). At first sight, this might provide some skepticism regarding the credibility of the heuristic. However, White (2001) states that the sole purpose in using the sample variance is to estimate the homogeneity of the truncated series. In other words, the *MSER* tries to observe the *behavior* of the standard error estimate, and detect the truncation point from this behavior. The underlying assumption—which is not explicitly stated by White—is that the behavior of the s.e. will approximately remain the same—regardless of autocorrelation in the sequence. The second problem is a practical one: the technique is very sensitive to outliers (extreme values). For instance, the sequence used in Fig. 1 contains eight extreme data points among which the smallest one is approximately 43 times larger than the mean of the sequence. We have observed in Fig. 1(a) that cumulative averages plot was not affected much by the existence of these outliers. However, the *MSER*-5 applied to the whole sequence suggests truncating 4876 observations, whereas deleting these extreme values from the sequence would change the truncation point drastically to 339. This shows that unless extreme values are carefully deleted from a sequence, *MSER* can display a poor performance.

4. Experimental design

We consider two types of manufacturing systems: (1) serial production lines and (2) job-shops. Both types are extensively studied in the literature (see Dallery and Gershwin, 1992).

4.1. Serial production lines

Fig. 2 shows a typical serial production line. The system consists of N serially arranged machines M_i, $i = 1, 2, \ldots, N$, with buffers B_i, $i = 1, 2, \ldots, N-1$, between two consecutive machines.

This system is an asynchronous, saturated system with machines having mutually independent processing times. Each machine can process at most one unit at a time, and has an internal storage capacity for that unit. All buffers in the system have finite storage capacities. Hence, blockages and starvation may occur; however, the first machine never gets starved, and the last machine never gets blocked. Machines are subject to random failures with independent inter-failure and repair times. No reworks or scraps are allowed. There is only one type of product; it visits all the N machines in the system in the given sequence. We assume empty and idle initial conditions in the simulation of this system.

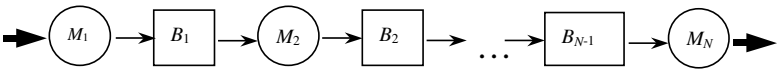

Fig. 2. N-staged serial production line.

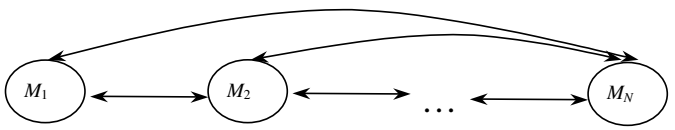

Fig. 3. N-machine job-shop production system.

Table 2
Experimental factors and levels for the serial-line system

Factors	Levels
System size	3, 9
Load type	Uniform, bottleneck (10%), bottleneck (20%), bottleneck (99%)
Load level	1, 0.9, 0.5
Processing time coefficient of variation	0.3, 2.5
Processing time variance	0.3, 2.5
Machine type	No-breakdown, unreliable (90% availability, FBSR[a]), unreliable (90% availability, RBLR[b]), unreliable (80% availability, FBSR), unreliable (80% availability, RBLR), unreliable (50% availability, FBSR), unreliable (50% availability, RBLR)
Buffer capacity	0, 10, 100

[a] FBSR: Frequent breakdown short repair time.
[b] RBLR: Rare breakdown long repair time.

Table 3
Experimental factors and levels for the job-shop system

Factors	Levels
System size	3, 9
Load type	Uniform, bottleneck (5%), bottleneck (10%)
Load level	80%, 50%
Processing time coefficient of variation	0.3, 1.0
Processing time variance	0.3, 1.0
Machine type	No-breakdown, unreliable (90% availability, FBSR[a]), unreliable (90% availability, RBLR[b])

[a] FBSR: Frequent breakdown short repair time.
[b] RBLR: Rare breakdown long repair time.

4.2. Job-shop production system

Fig. 3 shows a typical job-shop. This system shares many characteristics with serial-lines. The difference is that it has no intermediate storage buffers. A part still must visit all the machines. However, its processing sequence is not known in advance, but is determined randomly. Each part can visit each machine exactly once; each machine is equally likely to be selected in the sequence. The arrival pattern of parts to the system is a Poisson process; hence every machine in this system is allowed to starve. A newly arrived part waits in the system, until the first machine in its processing sequence becomes available for processing.

4.3. Experimental factors

Tables 2 and 3 present the experimental factors and their levels for our serial-lines and job-shops. Lognormal distribution (a continuous skewed distribution) is chosen to represent the processing times of machines, as is often the case in practice (Law and Kelton, 2000, p. 678). When experimenting with unreliable machines, we assume the uptime and downtime distributions to be gamma with shape parameter 0.7 and 1.4, as suggested by Law and Kelton (2000, pp. 681–682). The scale parameters are then calculated as discussed in their book. We now discuss the factors and their levels.

System size: Number of machines in the system. It has two levels for both systems.

Load type: It is the distribution of the total workload of the system across the machines. If we have $2n + 1$ machines in our system and the total workload is K time units per job, then

- for *uniform load type*, every machine works $(2n + 1)/K$ time units on each job (on the average),

- for x% *bottleneck load type*, x% of the uniform work times of the first and last n machines is transferred to $(n+1)$th machine, so that $(n+1)$th machine becomes the bottleneck machine of the system.

For instance, consider a 3-stage serial line with a total workload of 3 minutes per job. For the uniform version of this system, we split the total workload evenly between the machines, so that it takes 1 minute in each of the three machines to process the job. For the 10% bottleneck version of this system, the average processing time of a job in the first and the third machines is 0.9 $(=1-0.1\times 1)$ minutes, and the average processing time of a job in the second machine is 1.2 $(=1+0.1\times 1+0.1\times 1)$ minutes.

In a way, *load type* also determines if there exists a bottleneck in the system. Only one machine is allowed to be the bottleneck; it is always the machine that is in the middle of the part's processing sequence. The total workload of the system is kept constant; only the distribution of loads among machines is changed as discussed above. The magnitude of bottleneck is also investigated by changing its level from 10% to 20% to 99%. Since the simulation of job-shops requires considerable amount of runtime, the magnitude of bottleneck is kept small (5% and 10%) for these system. This factor has four and three levels for serial-lines and job-shops, respectively.

Load level: Average amount of work load in the system. For serial-lines, it has three levels, and is adjusted by the mean processing times of machines. Smaller values indicate highly loaded systems. For job-shops, it has two levels due to the extensive runtime requirements; it is adjusted by the arrival rate of parts. Larger values indicate highly loaded systems.

We distinguish between two types of variability measures; namely, the processing time's variance (PV) and coefficient of variation (CV), because problems would arise in interpreting the results for bottleneck systems. If the PV is kept constant, then the non-bottleneck machines will have higher CV, whereas the bottleneck machine will have lower CV than their uniform counterparts. Similar arguments can be given for the constant CV case.

Processing time coefficient of variation (CV): It has a low and high level as is usually done in related studies (see, Erel et al., 1996). The high level for the job-shop is chosen to be 1.0 instead of 2.5 due to long runtimes.

Processing time variance (PV): It has the same levels as CV.

Machine type: It is the reliability of each machine. Besides reliability itself, its magnitude is also investigated; we choose three levels for the long-run availabilities of machines for the serial-lines. But, due to long runtimes, only one availability level is chosen for the job-shops. A further aspect, the type of unreliability is also studied. Hopp and Spearman (2000) show that—given the same availabilities—a system experiencing frequent breakdowns but short repair times is preferable to a system experiencing rare breakdowns but long repair times. Thus there are seven levels for the serial-lines and three levels for job-shops. Table 4 lists the parameter levels used for reliability.

Buffer capacity: This factor is investigated for serial-lines only because of the no intermediate buffer assumption in job-shops. It has three levels, which are chosen considering the analytical results found in Conway et al. (1987).

5. Results of simulation experiments

We start this section by explaining the syntax and the structure used to present a large number of results. Only a representative set of results will be shown here due to space considerations (for detailed results, see Sandikci and Sabuncuoglu, 2004). Both the cumulative averages plot and the $MSER$ output are given in a single figure. The x-axis of each figure is the number of data truncated, whereas the y-axis is the time-in-system statistic. Each figure includes three different plots, corresponding to the cumulative averages plots for different designs. Each design is indicated by a specific name, which is written close to the associated plot. The numbers in parentheses, next to the design names, indicate the truncation points according to $MSER$.

Table 5 explains the meaning of the associated names for the serial-lines. For instance, design

Table 4
Breakdown scenarios

Availability	MTBF[a]	MRT[b]	TST[c]	Breakdown type
90%	9	1	10	Frequent breakdown short repair time
	90	10	100	Rare breakdown long repair time
80%	8	2	10	Frequent breakdown short repair time
	80	20	100	Rare breakdown long repair time
50%	5	5	10	Frequent breakdown short repair time
	50	50	100	Rare breakdown long repair time

[a] MTBF: Mean time between failures (in hours).
[b] MRT: Mean repair time (in hours).
[c] TST: Total system time (in hours).

Table 5
Design codes for serial production lines with no breakdowns

System size	Proc. time dist.	Variability	Workload	Dummy
'3' = 3 machines	'1' = Lognormal	'1' = 0.3 (CV)	'1' = uniform(1)	'1' = 0
'9' = 9 machines		'2' = 2.5 (CV)	'2' = bottleneck(1, 10%)	'2' = 10
		'6' = 0.3 (PV)	'3' = bottleneck(1, 20%)	'4' = 100
		'7' = 2.5 (PV)	'4' = uniform(0.9)	
			'5' = bottleneck(0.9, 10%)	
			'6' = bottleneck(0.9, 20%)	
			'7' = uniform(0.5)	
			'8' = bottleneck(0.5, 10%)	
			'9' = bottleneck(0.5, 20%)	
			'a' = bottleneck(1, 99%)	
			'b' = bottleneck(0.9, 99%)	
			'c' = bottleneck(0.5, 99%)	

Table 6
Unreliable design codes for both serial line and job-shop experiments

Design	Avail.	Uptime dist.	Downtime dist.	Breakdown type
xxxxx1221	90%	Gamma	Gamma	FBSR[a]
xxxxx1222	90%	Gamma	Gamma	RBLR[b]
xxxxx1223	80%	Gamma	Gamma	FBSR
xxxxx1224	80%	Gamma	Gamma	RBLR
xxxxx1227	50%	Gamma	Gamma	FBSR
xxxxx1228	50%	Gamma	Gamma	RBLR

[a] FBSR: Frequent breakdown short repair time.
[b] RBLR: Rare breakdown long repair time.

`3l224` in serial lines (see top plot in Fig. 4(b)) corresponds to the 3-machine serial-line having lognormal processing time distributions with a CV of 2.5, a 10% bottleneck with a workload of 3 minutes-per-job, a buffer capacity of 100, and no breakdowns.

In Table 6, we appended 4 digits to the previous design names to identify the unreliable versions (systems with breakdowns). For instance, the unreliable version of design `3l224` in serial lines, which is 90% available with frequent breakdowns but short repair times, is named as `3l2241221`.

5.1. Results for serial production lines

5.1.1. Buffer capacity

The results show that *increasing buffer capacity increases the length of the transient period* (see Fig. 4(a), (b), and (p)). This is an interesting result since buffers usually have positive affects on performance measures. A system with more buffer spaces typically needs more time to fill-up. As an example, consider Fig. 4(b). The cumulative averages plots suggest the transient period as the 2000, 2500, and 4000 observations for designs `3l221`, `3l222`, and `3l224`, respectively. The buffer

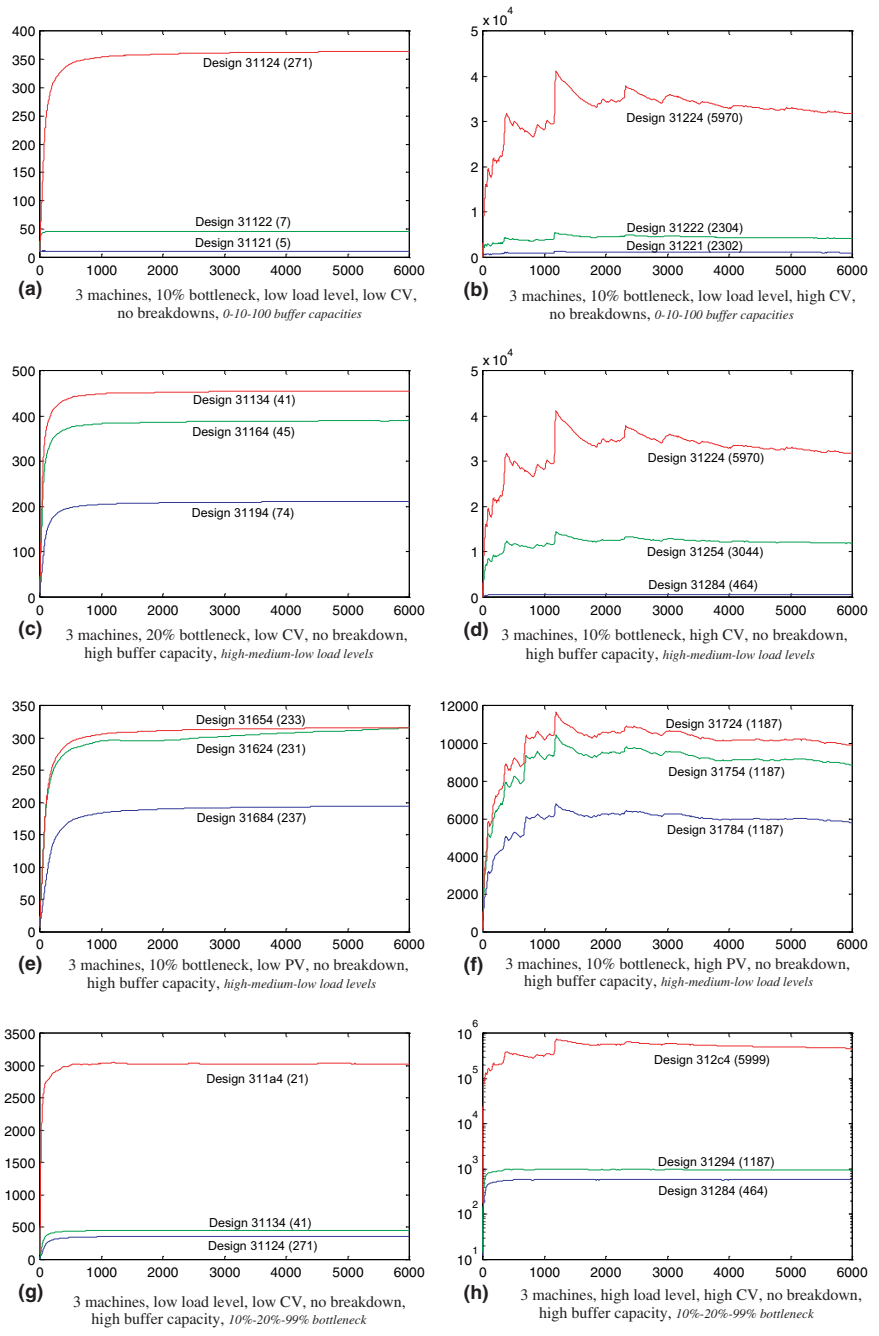

Fig. 4. Experimental results for serial lines; the numbers in parentheses are truncation points according to *MSER*; the differing parameters are indicated in *italics* in each figure; the three levels of these parameters represent the plots in a bottom to up fashion—e.g., in (a) the buffer capacity in designs `31121` (bottom plot), `31122` (middle plot), and `31124` (top plot) are 0, 10 and 100, respectively.

capacities in these designs increase from 0 to 10 to 100, respectively. The truncation points found by *MSER* for these designs are 2302, 2304, and 5970, respectively, which also comply with the

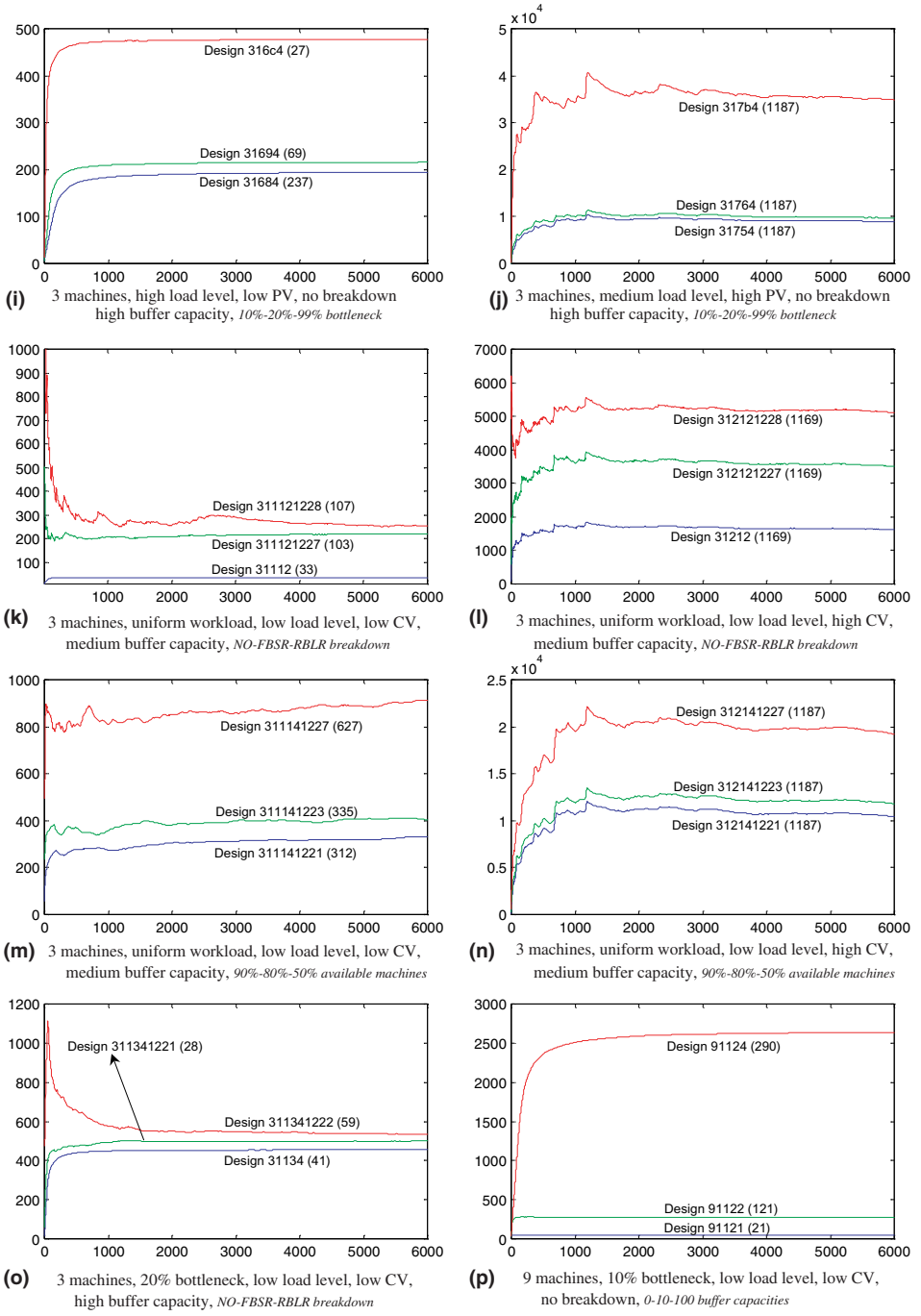

Fig. 4 (*continued*)

results of the cumulative averages plots. The same observation holds for other serial-line designs (see the many results in Sandikci and Sabuncuoglu, 2004).

5.1.2. Variability (CV, PV)

As expected, *increasing processing time variability measured by CV or PV significantly increases the length of the transient period*. The higher this variability, the higher the overall system variability is, the more coupling events between machines (i.e., interaction and interdependency between stations in terms of starvation and blocking), hence the longer the transient period. This effect can be viewed by comparing per row plots in Fig. 4(a) and (b), (e) and (f), (k) and (l), (m) and (n). Consider, for instance, Fig. 4(e) and (f). According to the cumulative averages plot, the system with a high load level in the low variable case (design `31684`) reaches steady-state at the 350th observation, whereas the corresponding system in the highly variable case (design `31784`) reaches steady-state at the 1000th observation. *MSER* results comply with these findings: truncate 237 and 1187 observations, respectively. The same behavior is observed in all other designs.

5.1.3. System size

Increasing system size significantly increases the length of the transient period (compare Fig. 4(a)–(p)). The design `31122` in Fig. 4(a), for instance, reaches steady-state at the 7th observation, whereas its counterpart in Fig. 4(p), i.e., design `91122`, reaches steady-state at the 121st observation. The same result is observed in all other designs.

This effect is mainly due to more coupling events in larger systems; it can also be explained by the following analogy. The process of achieving steady-state can be viewed as heating a building by several stoves. The length of the transient period is the time required to warm-up all the stoves to heat the entire building. The larger the building, the more stoves, hence the more energy or time is needed. Short lines resemble small buildings.

5.1.4. Load level

We begin this section with two observations:

Observation 1: "The buffers in a highly loaded system fill up faster since the system processes more parts per unit time. This causes a shorter transient period."

Observation 2: "The increase in load level causes an increase in the congestion level of the system, which results in more interactions among system entities, more coupling events, and more variability. And this causes a longer transient period."

Note that *MSER* results are more useful in making comparisons for this factor. The results are analyzed for two cases: low and high variability.

The results in the *low variability case* (either measured by CV or PV) indicate that *increasing load level increases the length of the transient period very slightly*. For the low CV case, we observe this effect by comparing the plots in Fig. 4(c). *MSER* suggests to truncate 41, 45, and 74 observations for designs `31134`, `31164`, and `31194`, respectively, which correspond to systems with a load level of 3, 2.7, and 1.5 minutes-per-job. The same behavior is observed for the low PV case, as shown in Fig. 4(e). Clearly, *Observation 2* outweighs *Observation 1* in the low variability case.

The results in the *high variability case* differ with respect to the type of variability measure. In the *case of high CV, the length of the transient period decreases significantly* as the load level decreases from 3 to 2.7 to 1.5 minutes/job (see Fig. 4(d)—truncation point decreases from 5970 to 3044 to 464, respectively). *Observation 1* shows its effect in this case. More importantly, increasing the load level causes an increase in the variability of the system via increased congestion, but a dominating decrease in the PV is attained since we kept CV constant. This is the main cause of the decrease of the transient period. However, in the *case of high PV, no change has been realized in the transient period with respect to load level* (see Fig. 4(f)). Keeping the variance constant at its high level (2.5) dominates every other effect in the system, so the same transient period results. Similar results hold for other designs.

5.1.5. Load type

We start this section with an example. A no-breakdown serial-line containing three machines with mean processing times given as 1-1-1 minutes/job is to be compared with its 99% bottleneck counterpart. To form the bottleneck, we need to transfer 99% of the work in the 1st and 3rd

machines to the 2nd machine, which produces a system with mean processing times given as 0.01–2.98–0.01 minutes/job. The processing times in the 1st and 3rd machines are small when compared with that of 2nd machine, so they may be neglected. This leads to the following.

Observation 3: "As the work is transferred to a single machine from other machines, the system can be viewed as getting smaller in size. Considering the results of Section 5.1.3, the length of the transient period is expected to decrease as the magnitude of the bottleneck is increased—given a constant workload system."

Again, the results are analyzed for two cases: low and high variability.

In the *case of low variability* (either measured by CV or PV), *the length of the transient period decreases with the increase in the magnitude of the bottleneck* (see Fig. 4(g) and (i) for low CV and low PV cases, respectively). For the low CV case, MSER suggest truncating 271, 41, and 21 observations as the magnitude of the bottleneck increases from 10% to 20% to 99% in these designs. Hence, the results are consistent with *Observation 3*. Remembering the stove analogy, we conclude that heating the biggest stove in the building is more important than heating the smaller ones to heat the entire building.

The results in the *high variability case* differ with respect to the type of variability measure. In the *case of high* CV (see Fig. 4(h)) results indicate a *significant increase in the length of transient period* (464 to 1187 to 5999) *with respect to the increase in the magnitude of bottleneck* (10% to 20% to 99%). Note that to keep CV constant, we increase the PV of the bottleneck machine. It was found in Section 5.1.2 that the increase in variability significantly increases the transient period. And it turns out that, in the case of high CV, the effect of variability dominates the effect discussed in *Observation 3*. However, in the *case of high* PV, *no change has been realized in the transient period with respect to load type* (see Fig. 4(j)). The variance (2.5) is high enough to compensate for any change in transient period that may be caused by the change in system size. Similar results are observed for other designs.

5.1.6. Machine type

Recall that this factor investigates the effect of: (i) the existence of unreliability, (ii) the magnitude of unreliability, and (iii) type of unreliability. The results in each category are given for two cases: low and high variability.

We start with the first category: *existence of unreliability*. In the *case of high variability* (either measured by CV or PV) *length of transient period is not affected by unreliable machines*; see Fig. 4(l). The following analogy would explain this result. The variability of a system can be viewed as the waves of a sea. A highly variable system resembles as a very wavy ocean. Hence waves that are generated by an artificial source will have no effect in the ocean unless the source is very powerful. By allowing the machines to breakdown, we are introducing additional variability to the system. However, the variability introduced by breakdowns is not much compared with the original variability of the system in the case of high CV. Hence, we do not observe any change in the transient period for high variability.

Fig. 4(k) shows that allowing breakdowns increases the length of the transient period in the low variability case. MSER suggests truncating 33 observations for the no-breakdown design (`31112`), whereas 103 and 107 observations are truncated for its 50% available frequent-breakdowns-short-repairs (FBSR) and rare-breakdowns-long-repairs (RBLR) counterparts (designs `3111212227` and `3111212228`, respectively). The same types of breakdown scenarios with 90% availabilities for design `31134` are shown in Fig. 4(o). This shows that the transient period increased only for the RBLR case, whereas it decreased for the FBSR case. Therefore, we conclude for the low variability case that the type and magnitude of unreliability have interacting effect on the transient period.

Next, we consider increasing the *magnitude of unreliability* from 90% availability to 80% and further to 50%. For the *high variability case*, the CV of processing times is the dominant factor—as discussed earlier (see Fig. 4(n)). Hence, there is *no change in the length of the transient period*. However, in the *low variability case*, there is *an increase in the length of transient period* as we move from

Table 7
Effect of utilization on the length of the transient period

Design	Average ρ[a]	Length of T_p[b]	Change in ρ	Change in T_p
3l12l	0.753	5	–	–
3l124	0.814	271	Increase	Increase
3l181	0.864	5	Increase	No change
3l221	0.346	2302	Decrease	Increase
3l1a1	0.356	2	Decrease	Decrease

[a] Utilization.
[b] Transient period.

90% available to 50% available ones, which is due to the increase in the variability introduced by breakdowns. That is, the more breakdown events occur, the higher variability is.

Finally, we consider the *type of breakdowns*. The results indicate that in the *high variability case* there is *no change in the length of the transient period* for FBSR and RBLR (designs 312121227 and 312121228 in Fig. 4(l)). However, for the *low variability case*, the results indicate that *rare but long breakdowns attain a longer transient period than frequent but short breakdowns* (see Fig. 4(k) and (o)).

5.1.7. Utilization

Although not previously listed among the experimental factors, we also studied the relationship between the length of the transient period and utilization of the system. The results indicated that *there is no direct relation* between these two measures. In some cases it increases, whereas it decreases in other cases (Table 7).

5.2. Results for job-shops

Since many of the results comply with the serial-line results, we will not give any figures for these systems; we shortly state the major results (for details see Sandikci and Sabuncuoglu, 2004): (i) increasing variability of the processing times significantly increases the length of the transient period, (ii) increasing system size increases the length of transient period, (iii) increasing load level causes a significant increase in the length of the transient period, (iv) introducing bottleneck machines increases the length of the transient period provided constant workload, (v) allowing breakdowns increases the length of the transient period, (vi) frequent but short breakdowns attain shorter transient period than rare but long breakdowns.

Although not listed among the experimental factors, we also analyzed the effect of finite buffer capacities in job-shops by relaxing the no intermediate buffer assumption and putting capacitated buffers with capacities of 10. The results indicate that systems with finite buffer capacities attain longer transient period than those with infinite buffer capacities.

6. Conclusions

In this paper, we studied the behavior of the initial transient period for non-terminating simulations of serial production lines and job-shops. We present the following conclusions and recommendations:

(1) As the variability of processing times increases, the transient period also increases—both for serial-lines and job-shops. In fact, variability is the most significant factor. If a particular system has highly variable processing times (i.e., CV $\geqslant$ 1), then the analyst should make fairly long runs to obtain enough observations from the steady-state distribution. We recommend running simulations long enough so that the ratio of the length of the transient period to the total run length does not exceed 25%.
(2) Increasing the system size increases the length of transient period. In our experiments, the system size is changed by changing the number of machines.
(3) The system load level has complicated effects on the transient period. For job-shops, increasing the load of the system increases the length of the transient period. For serial-lines, it increases the transient period only in the case of low variability, but it does so only slightly. However, the behavior changes for high variability cases. The transient period decreases in the high CV case, whereas there is no change in the high PV case.

(4) The load type has complicated effects. In job-shops, forming bottleneck machines and further increasing the magnitude of the bottleneck simply increases the length of the transient period. However, in serial-lines, introducing a bottleneck increases the transient period only in the high CV case (no change in the high PV case). In the low variability case (either low CV or low PV), the transient period decreases with increasing the magnitude of bottleneck, but only slightly.
(5) The existence of unreliable machines in a job-shop increases the length of transient period. In highly variable serial-lines, however, the transient period is neither affected by the existence of unreliable machines nor by the magnitude and type of unreliability. For the low variable serial-lines, the type and magnitude of breakdowns turns out to be more effective than just the existence of breakdowns. Increasing the magnitude of unreliability increases the transient period. Moreover, rare but long breakdowns cause a longer transient period than frequent but short breakdowns.
(6) The transient period increases with increased buffer capacities in serial lines, and with the introduction of capacitated buffers in job-shops.

A system having more variable output sequences will clearly have longer transient periods. Thus, simulation analysts should first investigate the change in the variability of output sequences. If any of the system's factors are suspected to introduce additional variability into a system, then a longer transient period should be expected. For instance, including unreliable machines in a system increases variability; however, this increase depends on the magnitude and type of unreliability. If alternative designs show similar variability but one of them has more entities than the other (e.g., more machines, or complicated material handling systems, etc.), then the analysts should base their decision about the length of transient period on the system with more entities. The degree of coupling in manufacturing simulations is an important factor that affects the transient period.

We also observed that, in most cases, both cumulative average plots and the *MSER* results are comparable. Cumulative averages usually suggest longer transient periods than *MSER*. Since the *MSER* is an objective criterion that yields results complying with one of the most frequently used graphical techniques, and is very simple and computationally efficient, we recommend this heuristic. However, special attention must be paid to remove any outliers from the sequence, which otherwise would lead the analysts to wrong conclusions. Moreover, it would be preferable—if there is enough time—to use both techniques.

A possible direction for future research is the study of the transient period in more complicated manufacturing simulations (e.g., automated-guided vehicles (AGVs), automated storage-retrieval systems (AS/RSs), etc.) and non-manufacturing simulations. It would be very useful for simulation practitioners if researchers could come up with an analytical expression that asks the user to enter system specific parameter values, which then gives the length of transient period. However, the authors have very little hope that this will happen.

References

Asmussen, S., Glynn, P.W., Thorisson, H., 1992. Stationarity detection in the initial transient problem. ACM Transactions on Modeling and Computer Simulation 2, 130–157.

Automod User's Manual v.9.0., AutoSimulation, Inc., Utah, 1999.

Cash, C.R., Nelson, B.L., Long, J.M., Dippold, D.G., Pollard, W.P., 1992. Evaluation of tests for initial-condition bias. In: Swain, J.J., Goldsman, D., Crain, R.C., Wilson, J.R. (Eds.), Proceedings of the 1992 Winter Simulation Conference, pp. 577–585.

Chance, F., 1993. A historical review of the initial transient problem in discrete-event simulation literature, Technical Report, School of Operations Research and Industrial Engineering, Cornell University.

Conway, R.W., Maxwell, W., McClain, J.O., Thomas, L.J., 1987. The role of work-in-process inventory in serial production lines. Operations Research 36 (2), 229–241.

Dallery, Y., Gershwin, S.B., 1992. Manufacturing flow line systems: A review of models and analytical results. Queuing Systems 12, 3–94.

Deligönül, Z.S., 1987. Antithetic bias reduction for discrete-event simulations. Journal of the Operational Research Society 38 (5), 431–437.

Erel, E., Sabuncuoglu, I., Kok, A.G., 1996. Analysis of serial production line systems for inter-departure time variability and average inventory, Technical Report IEOR-9618, Department of Industrial Engineering, Bilkent University.

Fishman, G.S., 1972. Bias considerations in simulation experiments. Operations Research 20, 785–790.

Fishman, G.S., 2001. Discrete Event Simulation: Modeling, Programming, and Analysis. Springer-Verlag.

Gafarian, A.V., Ancker Jr., C.J., Morisaku, T., 1978. The problem of initial transient in digital computer simulation. In: Winter Simulation Conference Proceedings, pp. 49–51.

Gallagher, M.A., Bauer Jr., K.W., Maybeck, P.S., 1996. Initial data truncation for univariate output of discrete-event simulations using the Kalman filter. Management Science 42 (4), 559–575.

Goldsman, D., Schruben, L.W., Swain, J.J., 1994. Tests for transient means in simulated time series. Naval Research Logistics 41, 171–187.

Harpell, J.L., Lane, M.S., Mansour, A.H., 1989. Operation research in practice: A longitudinal study. Interfaces 19 (3), 65–74.

Hopp, W.J., Spearman, M.L., 2000. Factory Physics. McGraw-Hill.

Kelton, W.D., 1985. Transient exponential-Erlang queues and steady-state simulation. Communications of ACM 28, 741–749.

Kelton, W.D., 1989. Random initialization methods in simulation. IIE Transactions 21 (4), 355–367.

Kelton, W.D., Law, A.M., 1983. A new approach for dealing with the startup problem in discrete event simulation. Naval Research Logistics Quarterly 30, 641–658.

Kelton, W.D., Law, A.M., 1985. The transient behavior of the M/M/s queue, with implications for steady-state simulation. Operations Research 33, 378–396.

Kleijnen, J.P.C., 1984. Statistical analysis of steady-state simulations: Survey of recent progress. European Journal of Operational Research 17, 150–162.

Law, A.M., 1984. Statistical analysis of the simulation output data. Operations Research 31, 983–1029.

Law, A.M., Kelton, W.D., 2000. Simulation Modeling and Analysis, third ed. McGraw-Hill.

MATLAB Version 5 User's Guide, Prentice Hall, 1995.

Ma, X., Kochhar, A.K., 1993. A comparison study of two tests for detecting initialization bias in simulation output. Simulation, 94–101.

Madansky, A., 1976. Optimal initial conditions for a simulation problem. Operations Research 24 (3), 572–577.

Murray, J.R., 1988. Stochastic initialization in steady-state simulations, Ph.D. dissertation, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor.

Murray, J.R., Kelton, W.D., 1988a. The transient response of the $M/E_k/2$ queue and steady-state simulation. Computers and Operations Research 15, 357–367.

Murray, J.R., Kelton, W.D., 1988b. Initializing for bias reduction: Some analytical results. In: Abrams, M., Haigh, P., Comfort, J. (Eds.), Proceedings of the 1988 Winter Simulation Conference, pp. 546–548.

Nelson, B.L., 1990. Variance reduction in the presence of initial-bias. IIE Transactions 22, 340–350.

Nelson, B.L., 1992. Initial-condition bias. In: Salvendy, G. (Ed.), Handbook of Industrial Engineering, second ed. Wiley.

Sandikci, B., Sabuncuoglu I., 2004. The behavior of the transient period of non-terminating simulations: An experimental analysis, Working paper 01-04, Department of Industrial Engineering, Bilkent University.

Schruben, L.W., 1981. Control of initialization bias in multivariate simulation response. Communications of the ACM 24, 246–252.

Schruben, L.W., 1982. Detecting initialization bias in simulation output. Operations Research 30, 569–590.

Schruben, L.W., Singh, H., Tierney, L., 1983. Optimal test for initialization bias in simulation output. Operations Research 31, 1167–1178.

Spratt, S.C., 1998. Heuristics for the startup problem, M.Sc. Thesis, Department of Systems Engineering, University of Virginia.

Vassilacopoulus, G., 1989. Testing for initialization bias in simulation output. Simulation 52, 151–153.

Welch, P.D., 1982. A graphical approach to the initial transient problem in steady-state simulations. In: Proceedings of the 10th IMACS World Congress on Systems, Simulation, and Scientific Computation, Montreal, pp. 219–221.

White Jr., K.P., 1997. An effective truncation heuristic for bias reduction in simulation output. Simulation 69 (6), 323–334.

White Jr., K.P., 2001. Personal communication.

White Jr., K.P., Cobb, M.J., Spratt, S.C., 2000. A comparison of five steady-state truncation heuristics for simulation. In: Joines, J.A., Barton, R.R., Kang, K., Fishwick, P.A. (Eds.), Proceedings of the 2000 Winter Simulation Conference, pp. 755–760.

Wilson, J.R., Pritsker, A.A.B., 1978a. A survey of research on the simulation startup problem. Simulation 31, 55–58.

Wilson, J.R., Pritsker, A.A.B., 1978b. Evaluation of startup policies in simulation experiments. Simulation 31, 79–89.