AN ABSTRACT OF THE THESIS OF

Budiman Saleh for the degree of Master of Science in Industrial and Manufacturing Engineering presented on December 12, 1991.

Title : Steady-State Analysis in Simulation : An Application of Schriber's Truncation Rule to Complex Queueing Systems.

Abstract Approved : Redacted for Privacy .

Sabah Randhawa

The objective of many steady-state simulations is to study the behavior of a nonterminating system with a peak load of infinite duration. Due to the complexity of the system, the initial conditions of the system are often atypical that often requires the simulators to start the system with the empty and idle conditions. Consequently, deletion of some initial observations is required to reduce the initialization bias induced by atypical initial conditions.

This paper studies the application of Schriber's truncation rule to the complex queueing systems (specifically, the two-machine and three-machine tandem queueing system) and the effects of parameter selection (i.e. parameters batch size and time between observations) on performance measures. Based on the previous studies of Schriber's rule on the one-machine system, parameters batch count and tolerance are held constant.

Mean-squared error and half length are used as measures of accuracy and interval precision in comparing the results.

The results of both systems show that time between observations and batch size are significant parameters, and the recommendations for the two-machine system can be generalized for the three-machine system. Increasing the number of machines in the system from two to three requires a careful reduction in the value of time between observations. Besides, multiple replications should be used to minimize the extreme results in determining the steady-state mean number of entities and the truncation point.

STEADY-STATE ANALYSIS IN SIMULATION : AN APPLICATION OF SCHRIBER'S TRUNCATION RULE TO COMPLEX QUEUEING SYSTEMS

by

Budiman Saleh

A THESIS

Submitted to
Oregon State University

in partial fulfillment of
the requirements for the degree of

Master of Science

Completed December 12, 1991

Commencement June 1992

APPROVED :

Redacted for Privacy

Professor of Industrial and Manufacturing Engineering in charge of Major

Redacted for Privacy

Head of department of Industrial and Manufacturing Engineering

Redacted for Privacy

Dean of Graduate School

Date Thesis is presented December 12, 1991

Typed by Budiman Saleh for Budiman Saleh

ACKNOWLEDGEMENTS

I would like to thank my major professor Dr. Sabah Randhawa for his continuous support, guidance, and encouragement during my graduate work and the course of this research.

I would also like to thank to many individuals from PT. IPTN, Indonesian Aircraft Industries, for their continuous involvement in sponsoring my graduate study in here.

I am thankful to my parents and Oom & Tante Juwono for their love, support, and understanding in achieving my goals.

Finally, I would like to express my sincere appreciation to Cindy for her patience, love, and understanding.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Table		Page

STEADY-STATE ANALYSIS IN SIMULATION : AN APPLICATION OF SCHRIBER'S TRUNCATION RULE TO COMPLEX QUEUEING SYSTEMS

CHAPTER 1. INTRODUCTION

1.1 Transient Versus Steady State Simulation

The two basic types of simulations with regard to analysis of the output data are transient (or terminating) simulation and steady state (or nonterminating) simulation. A transient simulation has a specified interval of simulated time $[0, T_E]$ in which the desired measures of system performance are observed. It means that the simulation begins at time 0 under certain initial condition(s) and ends when a specified event (or set of events) E occurs at time T_E. In this type of simulation, event (or set of events) E is defined by the nature of the simulated problem. It should be pointed out that the initial condition(s) and event (set of events) E must be well specified before the simulation begins.

On the other hand, a steady state simulation runs continuously as the length of simulation time goes to infinity (that is, for a long period of time). Since there is no natural event E defined by the nature of the problem to terminate the simulation, the simulator must decide when to stop the simulation - that is, after some number of observations have been collected and/or after length of time T_E has passed. Again, the initial condition(s) and event (set of events) E must be specified before the simulation begins.

Usually, a steady state simulation is used to study how the system will respond to a peak load of infinite duration. Thus, selecting the simulation type depends on what the simulator wants to learn about the system; see Law [1983].

1.2 Stochastic Nature of Simulation Analysis

Generally, there are three basic requirements that have to be satisfied in applying statistical inference methods or classical statistics to analyze the output data :

a. Observations are independent.

b. Observations are sampled from identical distribution.

c. Observations are drawn from a normal population.

However, the output data of the simulation experiment do not satisfy these requirements as explained below.

1.2.1 No independency.

In simulation, the output data from a discrete time stochastic process can be defined as the waiting time of the i^{th} customer or as the number of customers in a system that are sampled at equidistant time interval. Since the output data can be recorded over period of time, the data can be represented by the time series $\{X_1, X_2, \ldots, X_n\}$. In this study, X_i is the number of entities in the system observed at the i^{th} time.

The nature of time series data is such that the value of X_i may influence the value of its successor X_{i+1}. This means that the time series data are autocorrelated or $\{X_1, X_2, \ldots,$

$X_n\}$ are dependent. The sample mean

$$\overline{X}_{(n)} = \frac{1}{n} \sum_{i=1}^{n} X_i \tag{1.1}$$

remains an unbiased estimator for population mean μ_x; however, because of autocorrelation, the sample variance

$$s^2_{(n)} = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \overline{X}_{(n)})^2 \tag{1.2}$$

becomes a biased estimator for population variance σ^2.

The correlation between any two observations at lag-i (that is, i observations apart) is given by ρ_i and since the output data of most queueing simulations are usually positively correlated ($\rho_i > 0$, i = 1, 2, ..., n-1), the sample variance of the data $s^2_{(n)}$ underestimates the population variance σ^2 (see Banks and Carson [1984]), or :

$$\mathbf{E}[s^2_{(n)}] < \sigma^2 \tag{1.3}$$

1.2.2 No stationarity.

The output data $\{X_1, X_2, ..., X_n\}$ are recorded from a covariance stationary process only if the sample mean $\overline{X}_{(n)}$ and variance $s^2_{(n)}$ of the random variables (r.v.'s) X_i are stationary over time and the covariance between X_j and X_{j+i} (or Cov $[X_j, X_{j+i}]$) depends only on the separation i (i observations apart) and not on the actual time values of j and j+i.

However, the simulation output data are never strictly covariance stationary in practice because these data are usually sampled from two periods or phases, warm-up period and steady-state period. In more familiar terms, warm-up period is also known as transient period where no stationarity occurs. On the other hand, stationarity will occur when the system is already in the steady-state period and, then, the output data becomes time-independent.

The absence of stationarity implies that $\overline{X}_{(n)}$ and $s^2_{(n)}$ are not constant over time but always vary over time. Thus, the output data sampled from the warm-up period are also said to be time-dependent and the estimates may be biased and, hence, unreliable.

As mentioned earlier, the initial condition(s) must be well specified before the simulation begins. Wilson and Pritsker [1978b] found that the selection of initial condition(s) is more effective and has a greater influence on the accuracy of the performance measures than any other factor. The steady-state mode (or values close to steady-state mode) is found to be the best initial condition as it minimizes the warm-up period and prevents the discarding of too many observations. Since often little information or knowledge about the behavior of the system is known, the empty, idle condition is usually selected as the initial condition even though this selection can cause the output data sampled from the transient period to be significantly biased.

Schruben [1982] recommends that the output data be grouped into small adjacent batches (that is, five observations per batch) and that the sequence of batch means be compared in detecting the initialization bias. Because of this initialization bias, it is also recommended that the output data sampled from the transient period be discarded and that the sample mean be estimated based on the output data sampled from the steady-state period. Furthermore, Schruben [1983] suggests that the duration run or the number of observations per replication be increased if the output data still shows the initialization bias.

1.2.3 No normality.

The normality requirement can be relaxed by using the well known approach, the Central Limit Theorem. Therefore, it is important to know how to collect observations of independent and identically distributed (i.i.d) r.v.'s having population mean μ_x before applying the classical statistic methods.

1.3 **Truncation in Steady-state Simulation**

In order to determine the truncation point, Wilson and Pritsker [1978a] identified three common approaches : time series analysis techniques (by Fishman), queueing theory models (by Blomqvist, Cheng, Law, and Madansky), and heuristic rules (by Conway, Fishman, Gordon, and Schriber). Even though the results of time series analysis and queueing theory models are rigorous and precise, these two approaches have rather limited applicability because of the number of analytical parameters that have to be estimated or calculated before the simulation begins. As the system becomes more complex, the analytical parameters as well as the complexity of computations increase. In the worst case, these approaches may no longer fully describe the system. Also, to use these approaches, the practitioners are required to have a good background in time series analysis techniques or queueing theory models.

Heuristic rules are by far the more commonly used procedure for truncation. Their application is relatively simpler; hence greater acceptance by simulation practitioners. However, application of heuristic rules is still not straight forward. Usually the heuristic rules are ambiguously defined and require that certain statistical parameters be estimated or selected by the simulator before they can be applied. Thus, the application of the rules still depends on the judgement of the simulator or the analyst. In this study, only heuristic rules will be evaluated.

Gafarian et al [1978] evaluated the first comprehensive analysis of simulation startup policies used to identify the minimum truncation point such that the variation between the sample mean $\overline{X}_{(n,d)}$ and population mean μ_x is within the preassigned tolerance or controllable limit ϵ. It means that a simulator must know the conditions of the simulated model at the time the data are collected - either from periods of transient or steady state - so that excessive truncation or lack of data can be avoided.

To compare the performance of the heuristic truncation rules, Gafarian et al [1978] developed a set of criteria consisting of accuracy, precision, generality, cost, and simplicity. In their research, they did not examine the full effects of random variation of truncation point on the sample mean $\overline{X}_{(n,d)}$ as an estimator of population mean μ_x. Besides, the results show that the best policy for estimating μ_x may not necessarily also be the best policy for estimating the minimum truncation point.

It has been suggested in many heuristic rules to delete the data collected from the transient period and to calculate the sample mean based on the data collected from the steady-state period; see Schruben [1982, 1983]. The sample mean is now known as the truncated sample mean. This truncated sample mean

$$\overline{X}_{(n,d)} = \frac{1}{n-d} \sum_{i=d+1}^{n} X_i \tag{1.4}$$

where :

n = number of observations

d = number of data to be deleted

was used by Fishman [1972] in anlyzing the effects of initial conditions on a first-order autoregressive process (a special case of time series analysis techniques).

Fishman reported that deleting the first d observations reduces the bias and increases the variances because of the loss of data or information in a fixed sample size simulation. However, deleting more observations - that is, increasing the truncation point - increases the mean-squared error of the sample mean according to the following equation :

$$MSE_{(\overline{X}_{(n,d)})} = Var[\overline{X}_{(n,d)}] + (\mathbf{E}[\overline{X}_{(n,d)}] - \mu_x)^2 \qquad (1.5)$$

Fishman concluded that deletion of some observations is not always desirable since it worsens the variance. Thus, the bias reduction must be carefully weighed against the increased variance. This result was also supported by Turnquist and Sussman [1977].

As mentioned earlier, the selection of initial condition(s) has a greater influence on the accuracy of the measures of performance than any other factor, including the choice of truncation method. However, since the empty, idle condition is usually selected as the initial condition, Kelton and Law [1984] reported that replication of some independent runs and deletion of some initial observations is still an effective and efficient method of dealing with initialization bias.

To perform the heuristic truncation rules, the simulator must also know how to collect the output data which are autocorrelated. Methods used to collect the data and heuristic truncation rule applied in this study are explained below.

1.4 Output Data Collection

Besides minimizing the initialization bias, the analyst must consider how to minimize the autocorrelation effect found in the output data $\{X_1, X_2, \ldots, X_n\}$. Two general approaches that have been used in heuristic truncation rules to construct the unbiased estimator for population mean μ_x and variance σ^2 are fixed sample size approach and sequential approach.

In fixed sample size approach, one simulation run or several independent runs of an arbitrary fixed length or fixed number of observations are performed to construct the point estimate and a confidence interval (c.i.). However, in sequential approach the length of a single simulation run is sequentially increased until an acceptable c.i. can be constructed; as such, this method depends only on the availability of data. This study uses the fixed sample size approach.

In literature, six procedures using fixed sample size approach have been reported. These are : replication, batch means, spectrum analysis, autoregressive, regenerative, and standardized time series. From simulations of several queueing and inventory systems using coverage and half length as criteria for comparison, Law [1977] found that batch means method is superior to replication method although neither method worked well if the total sample size N is too small. Later in further research, Law [1983] found very little use of the methods of batch means, autoregressive, spectrum analysis, standardized time series, and regeneration cycles. The only method ever used because of its simplicity and familiarity is the replication method since it does not require the simulators or analysts to have a good statistical background; (see also Law and Kelton [1979, 1984]). Because of their applicability, only the methods of replication and batch means are discussed in sections 1.4.1 and 1.4.2, respectively.

1.4.1 Replication method.

- R independent simulation runs are performed with different random numbers for each run.
- In each run, n fixed observations will be recorded and the first d observations will be discarded due to the significant bias that occurs in the transient period.
- The truncated sample mean from a particular j^{th} run

$$\overline{X}_{j(n,d)} = \frac{1}{n-d} \sum_{i=d+1}^{n} X_{ij} \qquad \text{for } j = 1, 2, \ldots, R \qquad (1.6)$$

  is calculated based on the truncated (n - d) observations (Fishman [1972]). This truncated sample mean is considered as a single observation. The grand mean $\overline{\overline{X}}_{(R)}$ is then calculated as

$$\overline{\overline{X}}_{(R)} = \frac{1}{R} \sum_{j=1}^{R} \overline{X}_{j(n,d)} \qquad (1.7)$$

  as an estimator of μ_x.
- Advantage : Kelton and Law [1984] concluded that replication method can be a viable method of analysis in

steady state simulation because this method is very simple compared to other methods and n single observations are truly i.i.d unbiased observations.

- Disadvantage : Boundary effects due to autocorrelation still exist. Wasting data (or excessive truncation) due to some biased observations collected near the start of simulation run (i.e. transient period) cannot be avoided. Besides, each run starts with the same initial conditions that do not represent the steady-state behavior of the system being modeled.

1.4.2 Batch means method.

- One long simulation run is performed and the length of run is m fixed observations.
- The simulation period is divided into n batches and each batch mean

$$\overline{X}_j(b) = \frac{1}{b}\sum_{i=m-jb+1}^{m-(j-1)b} X_i \qquad , 1 \le j \le n \tag{1.8}$$

represents a single observation with a batch size of b. If the batch size b (b = m/n) is large enough, the batch means are approximately normal and uncorrelated.

- Advantage : One long run can dampen the initial effect of transient state so that the grand mean of batch means will be an unbiased estimator for μ_x.
- Disadvantage : The successive batch means may still reflect the boundary effects due to autocorrelation; this

may be more crucial than the replication method. It implies that the batch means will not exactly be from a covariance stationary process. Besides, it is difficult to identify the batch size b large enough so that the batch means follow an approximate normal distribution.

1.5 Truncation Rule

The heuristic truncation rule evaluated in this study is the Schriber's truncation rule. Schriber's truncation rule was chosen in this study because this rule is conceptually appealing. It uses batch means method to detect the initialization bias among the sequence of batch means and, then, applies replication method to calculate the truncated sample mean for each run. The advantages of using batch means method are that large enough batch size b will ensure practical independence of successive batch means and an adequate truncation point. Furthermore, using replication method will ensure that the truncated sample means from n runs are i.i.d observations. Also, the rule has previously been used in application to simple one-machine system (Baxter [1990]).

Wilson [1977] reported that the performance of Schriber's heuristic truncation rule was found to be consistent with other frequently cited truncation rules developed by Gordon and Fishman.

1.5.1 Schriber's truncation rule.

Schriber [1974] suggests that the approach to steady state operating conditions may be monitored by partitioning the observed time series $\{X_i : 1 \le i \le n\}$ into adjacent batches of some fixed size b. Then the behavior of the batch means can be used to determine whether the steady-state condition has been achieved in the k most recent batch means, that is $\{\overline{X_j}(b) : 1 \le j \le k\}$. This means that time series data from the k recent batches were already observed from the steady-state period and autocorrelation no longer occurred.

It is important to note that the extreme value in the set of batch means always occurs near the start of simulation run because of the selection of atypical initial condition, for example, empty and idle conditions. As simulation time elapses, the batch means as well as the time series data become relatively stable, i.e. convergence to steady-state conditions.

Although Schriber [1974] used a detailed example to illustrate this method, he actually selected a truncation point by "inspection" rather than by applying a specific algorithm to identify the appropriate truncation point. Wilson [1977] used a formulation of Schriber's truncation rule and specified the important parameters used in Schriber's rule. These are : batch size b, batch count k, and tolerance ϵ. Then the truncation point, d, is set at time n if the k most recent batch means all fall within the tolerance ϵ of each other :

$$\max \{|\overline{X}_j(b) - \overline{X}_l(b)| : 1 \leq j, l \leq k\} \leq \epsilon \qquad (1.9)$$

where :

$\overline{X}_j(b)$: the batch mean of the j^{th} batch

$\overline{X}_l(b)$: the batch mean of the l^{th} batch

Since the batch means are always compared in k pairs of batches of size b, the steady-state condition can only occur after time n = k x b. It means that the minimum truncation point is d_{min} = k x b and that it must be satisfied in each simulation run. If at that time, the truncation rule is satisfied then d = n. Otherwise, the oldest batch $\{X_1, X_2, \ldots, X_b\}$ is dropped and the batch mean for the next batch $\{X_{n+1}, X_{n+2}, \ldots, X_{n+b}\}$ is calculated. This procedure continues until the comparison of the most recent batch means yields the above condition.

It should be pointed out that the truncation point, d, is sensitive to the selection of parameters b, k, and ϵ. Thus, in order to use relatively small batch sizes that prevent an excessive truncation point but still ensure that no autocorrelation occurs among batch means, a simpler but less general approach to Schriber's rule was chosen. For a batch count of two with a batch size of b observed when the M/M/1/15 queue is in steady-state condition, the difference of $(\overline{X}_1(b) - \overline{X}_2(b))$ has an expected value of 0 or

$$E[\overline{X}_1(b) - \overline{X}_2(b)] = 0 \qquad (1.10)$$

and has approximately normal distribution. Then, the variance of the difference becomes

$$Var[\overline{X}_1(b) - \overline{X}_2(b)] = 2\ [\rho_0(b) - \rho_1(b)] \qquad (1.11)$$

where :

ρ_i(b) : autocovariance between two batch means separated by i batches.

$$\rho_i(b) = Cov\ [\overline{X}_j(b)\ ,\ \overline{X}_{j+i}(b)] \qquad (1.12)$$

The difference should not exceed the condition specified below

$$|\overline{X}_1(b) - \overline{X}_2(b)| \le Z_{\alpha/2} \cdot \sqrt{2[\rho_0(b) - \rho_1(b)]} \qquad (1.13)$$

or

$$|\overline{X}_1(b) - \overline{X}_2(b)| \le \epsilon \qquad (1.14)$$

After some experimentation, the evaluation of the above condition yielded the tolerance $\epsilon = 4.03$ at an α of 25%. Wilson and Pritsker's study [1978b] of Schriber's rule considered a single selection of batch count of two, batch size of five, and variable selection of truncation point with $\epsilon = 4.03$. However, they did not examine the effect of the length of equidistant time interval.

1.5.2 Application of Schriber's rule to a one-machine system.

Baxter [1990] used a Weibull distribution for service time and introduced a new parameter time scale T (average number of arrivals between observations) along with parameters b, k, and ϵ to study the effects of parameter selection using

Schriber's rule for one machine system (M/M/1/15). It should be noted that T/λ (where λ is average arrivals per time unit) represents time between observations (TBO), i.e. equidistant time interval. Mean number of entities in the system and mean-squared error (MSE) were selected as the performance measures.

Baxter's [1990] two-way ANOVA test results showed that only parameters b, T, and interaction between the effect of b and T were significant for the two dependent variables. The results show that as the time scale T increases, the time series data (number of entities in the system) become more consistent, that is convergence to steady-state condition. This research suggested to use T greater than 4.5 arrivals between observations or to use time interval greater than one time unit between observations.

Even though batch count k is not significant, it is suggested to use batch count of two instead of three because $k = 2$ is more sensitive, i.e., more rapidly detects when the steady-state condition has occurred. Besides, excessive truncation can be prevented.

Likewise, a batch size of five observations is more sensitive to detect the gradual changes in the number of entities in the system compared to a batch size of 10 observations.

1.6 Research Objectives

The objectives of this paper are to study the application of Schriber's truncation rule to more complex queueing models as well as to study the effects of parameter selection - parameters batch size (b) and time between observations (TBO) - on system performance measures. Following Baxter's [1990] results, the parameters batch count and tolerance will be held constant in this study. More specifically, the system studied in this research are the two-machine and three-machine tandem queueing system.

1.7 Performance Measures

Performance measures must be specified in order to describe the effects of parameter selection - batch size (b) and time between observations (TBO). The performance measures used in this study are :

- Mean number of entities in the system.
- Average MSE $(\overline{MSE})$.
- Number of initial observations with empty, idle system.
- Empirical truncation point distribution.

1.7.1 Mean number of entities in the system.

The random variable, observed time series data, $\{X_i : 1 \le i \le n\}$ represents the number of entities in the system observed at the i^{th} time. Then, the truncated sample mean from the j^{th} run is given by

$$\overline{X}_{j(n,d)} = \frac{1}{n-d} \sum_{i=d+1}^{n} X_{ij} \qquad \text{for } j = 1, 2, \ldots, R \quad (1.15)$$

where X_{ij} is the number of entities in the system observed at the i^{th} time from the j^{th} run.

Since each design level is run for R = 1000 runs, the grand mean or the overall mean

$$\overline{\overline{X}}_{(R)} = \frac{1}{R} \sum_{j=1}^{R} \overline{X}_{j(n,d)} \qquad (1.16)$$

becomes an unbiased estimator of the theoretical steady-state mean number of entities in the system, μ_x.

Furthermore, the estimated variance of the distribution of the sample values $\overline{X}_{j(n,d)}$'s is given by

$$S^2_{(R)} = \frac{1}{R-1} \sum_{j=1}^{R} (\overline{X}_{j(n,d)} - \overline{\overline{X}}_{(R)})^2 \qquad (1.17)$$

also known as the sample variance. The sample variance represents the variability within the grand mean $\overline{\overline{X}}_{(R)}$. Besides, the standard deviation or standard error (s.e.) of the distribution having $\overline{\overline{X}}_{(R)}$ as the grand mean is

$$s.e. = \sqrt{\frac{S^2_{(R)}}{R}} \qquad (1.18)$$

The bias in the point estimator $\overline{\overline{X}}_{(R)}$ is given by

$$B = \overline{\overline{X}}_{(R)} - \mu_x \qquad (1.19)$$

which represents the deviation between the grand mean $\overline{\overline{X}}_{(R)}$ and the population mean μ_x. In practice, it is desirable to have B as small as possible so that the point estimator is said to be unbiased.

In order to assess how close $\overline{\overline{X}}_{(R)}$ is to μ_x, mean-squared error (MSE) and half length (HL) will be used as measures of accuracy and precision. MSE encompasses both the bias and variance since MSE is the sum of the bias squared and variance. Thus, for each design level the mean-squared error becomes

$$MSE = B^2 + S^2{}_{(R)} \tag{1.20}$$

or

$$MSE = [\overline{\overline{X}}_{(R)} - \mu_x]^2 + S^2{}_{(R)} \tag{1.21}$$

Since half length is used as a measure of confidence interval precision, a smaller HL is desirable for each design level. Performing R = 1000 runs for each design level allows the use of the normal approximation Z_α as the substitute of t_α. Thus the half length for each design level is given by

$$HL = (Z_{\alpha/2})\ (s.e.) \tag{1.22}$$

1.7.2 Average MSE, $\overline{MSE}$.

It is often desired in simulation to estimate the theoretical mean-squared error as the average of the mean-squared errors calculated from run to run rather than as the summation of the bias squared and variance of $\overline{\overline{X}}_{(R)}$ (equation 1.21) for each design level. Thus, the random variable of interest from the j^{th} run is

$$Y_j = [\overline{X}_{j(n,d)} - \mu_x]^2 \tag{1.23}$$

so that

$$\mathbf{E}[Y_j] = \mathbf{E}[(\overline{X}_{j(n,d)} - \mu_x)^2]$$

$$\mathbf{E}[Y_j] = Var[\overline{X}_{j(n,d)}] + [\overline{X}_{j(n,d)} - \mu_x]^2 \qquad (1.24)$$

$Var[\overline{X}_{j(n,d)}]$ is also known as the estimated variance of the distribution of the sample values X_{ij}'s recorded from the j^{th} run such that

$$Var[\overline{X}_{j(n,d)}] = S^2_{j(n,d)} = \frac{1}{n-d-1} \sum_{i=d+1}^{n} [X_{ij} - \overline{X}_{j(n,d)}]^2$$

for $j = 1, 2, \ldots, R$ (1.25)

while the bias in the theoretical sample mean from the j^{th} run is given by

$$B_j = \overline{X}_{j(n,d)} - \mu_x \qquad (1.26)$$

Thus, combining equations 1.24 and 1.25 will give

$$\mathbf{E}[Y_j] = S^2_{j(n,d)} + [\overline{X}_{j(n,d)} - \mu_x]^2$$

$$\mathbf{E}[Y_j] = S^2_{j(n,d)} + B^2_j$$

or

$$\mathbf{E}[Y_j] = MSE_j \quad \text{for } j = 1, 2, \ldots, R \qquad (1.27)$$

Let the new random variable

$$\acute{Y}_j = MSE_j \quad \text{for } j = 1, 2, \ldots, R \qquad (1.28)$$

so that the average mean-squared error is given by

$$\overline{MSE} = \frac{1}{R} \sum_{j=1}^{R} MSE_j$$

$$\overline{MSE} = \frac{1}{R} \sum_{j=1}^{R} (S^2{}_{j(n,d)} + B^2{}_j)$$

or

$$\overline{MSE} = \overline{S^2} + \overline{B^2} \tag{1.29}$$

It should be noted that

$$\overline{S^2} = \frac{1}{R} \sum_{j=1}^{R} S^2{}_{j(n,d)} \tag{1.30}$$

and

$$\overline{B^2} = \frac{1}{R} \sum_{j=1}^{R} B^2{}_j \tag{1.31}$$

are the average variance and average squared bias, respectively. Thus, the average mean-squared error ($\overline{MSE}$) is an unbiased estimator of the theoretical mean-squared error. Besides, since the bias can be either positive or negative, the bias must be squared, then, averaged. Schwarz inequality (see Neuts [1973]) implies that

$$\overline{B^2} \le [\overline{B}]^2 \tag{1.32}$$

1.7.3 Number of initial observations with empty, idle system.

Number of initial observations is used to describe the effects of changing the parameter, time between observations (TBO), during the simulation run. The random variable of interest is I_j which represents number of occurances of the empty, idle system during the j^{th} simulation run. Thus, the sample mean becomes

$$\bar{I}_{(R)} = \frac{1}{R}\sum_{j=1}^{R} I_j \qquad \text{for } j = 1, 2, \ldots, R \qquad (1.33)$$

as the direct estimator of the population mean number of initial observations μ_I.

Furthermore, the estimated variance of the distribution of sample values I_j's is given by

$$S^2_{(R)} = \frac{1}{R-1}\sum_{j=1}^{R} [I_j - \bar{I}_{(R)}]^2 \qquad (1.34)$$

The standard deviation or standard error of the distribution having the sample mean $\bar{I}_{(R)}$ is the same as in equation 1.18 mentioned earlier.

1.7.4 Truncation point distribution.

The empirical truncation point distribution is used for comparison in this study in order to describe any significant difference caused by parameter selection.

The random variable of interest is d_j which represents the truncation point of the j^{th} run and the sample mean of the truncation point is given by

$$\bar{d}_{(R)} = \frac{1}{R}\sum_{j=1}^{R} d_j \qquad (1.35)$$

as an estimator of the population mean truncation point μ_d. The standard deviation (standard error) is given as in equation 1.18.

CHAPTER 2. APPLICATION TO A TWO-MACHINE SYSTEM

2.1 Analytical Model

The first queueing system evaluated is a two-machine system consisting of two M/M/1/15 models placed in tandem - also known as a system with two queues in tandem. The M/M/1/15 model is used in this study because it is the most commonly used, cited model in many literature for a single-machine system; see Gafarian et al [1978], Schruben [1982], Kelton and Law [1983], Schruben, Singh, and Tierney [1983], and Baxter [1990].

The arrival rate λ of 4.5 arrivals per time unit to the system (or 0.222 time units between arrivals) and service rate μ of five jobs per time unit (or 0.2 time units per job) for each machine are used in these models with the finite queue capacity of 15 per queue. Both the arrival and service rates are distributed exponentially.

The utilizations of the first and the second machines are $\rho_1 = 0.9$ and $\rho_2 = 0.877$, respectively, and the theoretical steady-state mean number of entities in the system μ_x is 10.2631. More details about analytical results are given in Appendix I.

2.2 Computer Model

All the computer programming was done using SIMAN and FORTRAN languages. The important subroutines that will be discussed below are subroutines ARIM1, M1TOM2, ENDOS, INCTIM, PAIRCOM, TRUNC, and SCSTAT, respectively. The flowcharts for the subroutines ARIM1, M1TOM2, and ENDOS are given in Appendices II-A, II-B, and II-C, respectively.

2.2.1 Subroutine ARIM1 (arrive and start processing on machine-1).

The functions of this subroutine are as follows :

- to schedule the next arrival according to the arrival rate which is distributed exponentially.
- to process the current job and schedule its completion time on machine-1 if this machine is idle. After processing is completed, the current job will be sent to machine-2.
- to make the job wait in the first queue if machine-1 is busy and a space is available in that queue.
- to make the job leave the system without service if machine-1 is busy and no space is available in the first queue.

2.2.2 Subroutine M1TOM2 (start processing on machine-2).

The functions of this subroutine are as follows :

- to process the job sent from machine-1 and schedule its completion time on machine-2 if machine-2 is idle. After the process is completed, the job will be sent to leave the system.

- to make the job wait in the second queue if machine-2 is busy and a space is available in that queue.
- to block machine-1 if machine-2 is still busy and no space is available in the second queue. If the blockage occurs, machine-1 will not process a new job while machine-2 keeps processing the job in progress.
- to process the waiting job in the first queue (if any) every time machine-1 is idle and, then, schedule its completion time.

2.2.3 Subroutine ENDOS (leave the system).

The functions of this subroutine are as follows :

- to process the waiting job in the second queue (if any) every time machine-2 is idle and, then, schedule its completion time to leave the system.
- to unblock machine-1 whenever there is an available space in the second queue by moving the blocking job from machine-1 to that queue.
- to process the waiting job in the first queue (if any) every time machine-1 is unblocked and idle. Then schedule its completion time on machine-1 and send the job to machine-2.

2.2.4 Subroutine INCTIM.

All 50 observations to record the number of entities in the system are performed at time TBO, 2 TBO, ..., 50 TBO by this subroutine for each simulation run. The value of TBO will be varied in this study.

2.2.5 Subroutine PAIRCOM (pairwise comparison).

PAIRCOM subroutine performs the following :

- The batch means for the first b observations $\{X_1, X_2, \ldots, X_b\}$ and the second b observations $\{X_{b+1}, X_{b+2}, \ldots, X_{2b}\}$ are calculated.
- If the difference in these means is less than or equal to the controllable limit $\epsilon = 4.03$ ($\alpha = 0.25$), it implies that the steady-state condition has been achieved.
- Otherwise, the observations are still collected from the transient period. Thus, the oldest batch $\{X_1, X_2, \ldots, X_b\}$ is dropped and the batch mean for the next new batch $\{X_{2b+1}, X_{2b+2}, \ldots, X_{3b}\}$ is calculated. Then, the comparison between the latest two batch means is again performed.
- The comparison process stops as soon as the difference is less than or equal to $\epsilon = 4.03$.

2.2.6 Subroutine TRUNC (determining the truncation point).

As soon as the comparison process stops, subroutine TRUNC will do the following :

- Determining the truncation point d that could be 2b, 3b, ..., or 50. If the truncation point is less than 50, the (50 - d) observations collected after the truncation point will be kept for further calculations. However, if the truncation point is 50, only the last b observations will be kept.

- Calculating the single steady-state mean $\overline{X}$ and the mean-squared error $\overline{MSE}$ for the remaining observations. The mean and mean-squared error are stored in a file and are considered a single observation of a particular run. Notice that there are 1000 replications or runs for each design level in this study.

2.2.7 Subroutine SCSTAT.

This subroutine calculates the overall steady-state mean, bias, variance, mean-squared error, standard error, and half length from all 1000 runs.

2.3 Experimental Design

Baxter [1990] in an earlier research studied the effects of parameter selection using Schriber's rule for one-machine system (M/M/1/15). The results show that the parameters, time scale T and batch size b, are significant parameters that affect the dependent variables (that is the steady-state mean and mean-squared error), while the parameter batch count k is not significant. Also, the interaction between parameters time scale T and batch size b is significant.

For the one-machine system, Baxter suggests to use time scale T greater than 4.5 arrivals per observation (that is, one time unit between observations) in order to yield consistent results such that recording initialization bias condition or empty system during observation can be minimized. Besides, even though batch count k is not significant, it is suggested to use k of two batches (instead of three batches) because k of two batches is more sensitive to detect the occurance of steady-state condition so that discarding of too many initial observations can be avoided. For practicality, batch size should be either five or 10 observations per batch; see Schruben [1982] and Schruben, Singh, and Tierney [1983].

In this study, the observation time is controlled by time between observations (TBO) instead of by time scale T (average arrivals per observation). Notice that the TBO is a ratio between time scale T and the arrival rate, $\lambda = 4.5$ arrivals per time unit.

The experimental design for a two-machine system is a 5 x 2 design; there are five levels for time between observations and two levels for batch size b, Table 2.1.

Table 2.1 : Parameters and factor levels (5 x 2) for a two-machine system's experimental design.

Parameters	Factor levels
T : time scale [average arrivals / observations]	1.125; 2.25; 3.375; 4.5; 5.625
TBO : time between observations [time units]	0.25; 0.5; 0.75; 1.0; 1.25
b : batch size [number of observations per batch]	5; 10

Based on Baxter's suggestions, the batch count, k, is held constant at two batches. For each design level, 1000 replications are run; this resulted in a total sample size of 10,000.

In order to minimize the variation in the simulation results and to yield more consistent results, the synchronization technique is applied to each design level. This technique enables to use the same sequence of random numbers and to let the corresponding replications start with the same seed number selected automatically during simulation run. Besides, to generate samples from the exponential distribution, the inverse transfom method is used, see Kelton and Law [1982]. Furthermore, different random number streams are used for arrival times and service times for the two machines, see Banks and Carsons [1984].

2.4 <u>Results</u>

Equations 1.16, 1.17, 1.18, 1.19, 1.21, and 1.22 are used to calculate the grand mean, sample variance, standard error (or standard deviation), bias, mean-squared error, and half length, respectively. A summary of the overall steady-state mean number of entities in the system, bias, variance, mean-squared error, standard error, and half legth (with α = 10%) for the 10 design levels is presented in Table 2.2.

The table shows that variance, mean-squared error, and half length are minimized with the parameter set of TBO = 0.25 time units and a batch size of five. The bias, however, is minimized with the parameter set of TBO = 0.5 time units and a batch size of five. It should be pointed out that since all 10 design levels are done with batch count of two, this means that two-sequential batch means are being compared. The following sections evaluate specific performance measures in more details.

Table 2.2 : Overall steady-state results for a two-machine system, with an α of 10%.

TBO, b	Mean	Bias	Var.	MSE	s.e.	HL
0.25, 5	7.546	-2.718	13.830	21.215	0.118	0.194
0.25,10	8.250	-2.013	21.052	25.106	0.145	0.239
0.5, 5	9.953	-0.311	26.212	26.308	0.162	0.266
0.5, 10	10.821	0.558	40.296	40.607	0.201	0.330
0.75, 5	11.655	1.392	35.145	37.084	0.186	0.308
0.75,10	12.435	2.172	50.131	54.848	0.224	0.368
1.0, 5	13.169	2.906	51.116	59.558	0.226	0.372
1.0, 10	14.286	4.023	73.204	89.386	0.271	0.445
1.25, 5	14.511	4.248	56.976	75.022	0.239	0.393
1.25,10	15.650	5.387	76.802	105.817	0.277	0.456

Note :

- λ = 4.5 arrivals per time unit.
- TBO : time between observations [time units].
- There are 1000 replications used in each design level to construct the single-point estimate, mean number of entities in the system.
- Bias represents the deviation between the overall mean and the theoretical steady-state number of entities (μ_x = 10.2631), while variance represents the variability within the overall mean. Finally, MSE is the summation of squared bias and variance.

2.4.1 Steady-state mean number of entities in the system.

Before the effects of parameter selection on the steady-state mean can be studied, the two-way ANOVA test on mean as the dependent variable is performed to test the significance of the main factors and factor interaction. The ANOVA results (Table 2.3) show that the main effects for the two parameters TBO and batch size are significant, while the interaction between these parameters is not significant.

Table 2.3 : ANOVA for a two-machine system with steady-state mean as the dependent variable.

Dependent variable : Mean					
Source	DF	Anova SS	Mean Square	F value	Pr > F
TBO	4	63515.502	15878.876	357.02	0.0001
Batch	1	2122.881	2122.881	47.73	0.0001
TBO*Batch	4	77.787	19.447	0.44	0.7818

After performing ANOVA test, Duncan's Multiple Range Test (DMRT) is performed to rank the design-level means statistically. DMRT is performed by creating a new variable COND (for "condition") that represents a combination of the original independent variables, TBO and batch size. The results of DMRT are as follows (Table 2.4) :

- As TBO increases for a fixed batch size, the mean number of entities increases.
- As the batch size increases for a fixed TBO, the mean number of entities increases.

Table 2.4 : Ranked steady-state means by using DMRT with α = 10%.

Duncan Grouping	Mean	N	COND
A	15.650	1000	1.25, 10
B	14.511	1000	1.25, 5
B	14.286	1000	1.0, 10
C	13.169	1000	1.0, 5
D	12.435	1000	0.75, 10
E	11.655	1000	0.75, 5
F	10.821	1000	0.5, 10
G	9.953	1000	0.5, 5
H	8.250	1000	0.25, 10
I	7.545	1000	0.25, 5

Means with the same letter are not different significantly.

- The means for parameter set of TBO = 1.25 with a batch size of five and for parameter set of TBO = 1.0 with a batch size of 10 are not significantly different, while means for other parameter sets are significantly different from each other.

2.4.2 Average mean-squared error ($\overline{MSE}$).

For each design level, the average squared bias, average variance, and average mean-squared error are calculated based on each single value of squared bias, variance, mean-squared error recorded from run to run using equations 1.31, 1.30, and 1.29, respectively. The single value of mean-squared error represents both bias and variance of a single mean number of entities from a particular run. A summary of these average values is presented in Table 2.5.

Table 2.5 : Average results for a two-machine system at an α of 10%.

TBO	Batch size b	Mean	$\overline{Bias^2}$	$\overline{Var}$	$\overline{MSE}$
0.25	5	7.5455	21.2014	11.8220	33.023
0.25	10	8.2497	25.0848	8.3495	33.434
0.5	5	10.8207	26.2825	20.8140	47.097
0.5	10	9.9526	40.5668	14.4893	55.056
0.75	5	12.4349	12.5640	23.4840	36.048
0.75	10	11.6554	51.7979	19.3657	71.164
1.0	5	14.2858	52.5070	34.9964	86.503
1.0	10	13.1686	73.3128	24.9181	97.231
1.25	5	15.6496	68.9652	39.2602	68.965
1.25	10	14.5112	93.7400	28.6432	93.740

The two-way ANOVA test and DMRT are performed on the average mean-squared error ($\overline{MSE}$) as the dependent variable for 10 design levels of independent variables TBO and batch size (Table 2.6). Table 2.6 shows that the main effects for the two parameters TBO and batch size as well as their interaction are significant. This means that it is not possible to study the effects of TBO or batch size on the average mean-squared error separately; the condition of both parameters must be known simultaneously to study the variations in $\overline{MSE}$.

Table 2.6 : ANOVA for a two-machine system with $\overline{MSE}$ as the dependent variable.

Dependent variable : $\overline{MSE}$					
Source	DF	Anova SS	Mean Square	F value	Pr > F
TBO	4	4567770.637	1141942.659	98.29	0.0001
Batch	1	623911.051	623911.051	53.70	0.0001
TBO*Batch	4	388826.850	97206.713	8.37	0.0001

The DMRT test at an α = 10% shows that (Table 2.7) :

- As the batch size increases for a fixed TBO, the average squared bias and the average mean-squared error increase. However, the average variance decreases.

Table 2.7 : Ranked $\overline{MSE}$'s by using DMRT with α = 10%.

Duncan Grouping	$\overline{MSE}$	N	COND
A	97.231	1000	1.0, 10
B A	93.740	1000	1.25, 10
B	86.503	1000	1.0, 5
C	71.164	1000	0.75, 10
C	68.965	1000	1.25, 5
D	55.056	1000	0.5, 10
D	47.097	1000	0.5, 5
E	36.048	1000	0.75, 5
E	33.434	1000	0.25, 10
E	33.023	1000	0.25, 5

$\overline{MSE}$'s with the same letter are not significantly different.

- As the TBO increases for a fixed batch size, the average squared bias, average variance, and average mean-squared error increase. Notice that Table 2.7 does not show the variability of the average mean-squared errors for a batch size of 10 and there is no significant difference in average mean-squared errors for TBO = 1.0 and TBO = 1.25 time units. However, the variability of the average mean-squared errors is found for a batch size of five such that there are significant differences in the average mean-squared errors for TBO = 0.25 and TBO = 0.5 time units as well as in the average mean-squared errors for TBO = 0.5 and TBO = 0.75 time units but no significant difference in the average mean-squared errors for TBO = 0.25 and TBO = 0.75 time units. In addition, the average mean-squared error for TBO = 1.0 is significantly higher than that of for TBO = 1.25 time units.

- The average mean-squared error is minimized with the parameter set of TBO = 0.25 time units and a batch size of either five or 10, because no significant difference is found in these values.

It is important to understand the cause of the variability of the average mean-squared errors for all design levels before making any conclusion about the effect of parameter selections on the average mean-squared error. This is discussed in detail in section 2.5.2.

2.4.3 Initial observations of empty, idle system.

In this study, 50 observations are recorded in each replication for each design level, the first observation being performed at simulation time 0.0 with the empty system as the initial condition. Then, the number of initial observations (that is, the occurance of the empty, idle system) is recorded for each replication. For illustration, Figures 2.1 and 2.2 show the first twenty observations taken from the first replication for different values of TBO. Figure 2.1 shows that there are at least 8, 1, and 1, observations of an empty, idle system corresponding to TBO values of 0.25, 0.5, and 0.75.

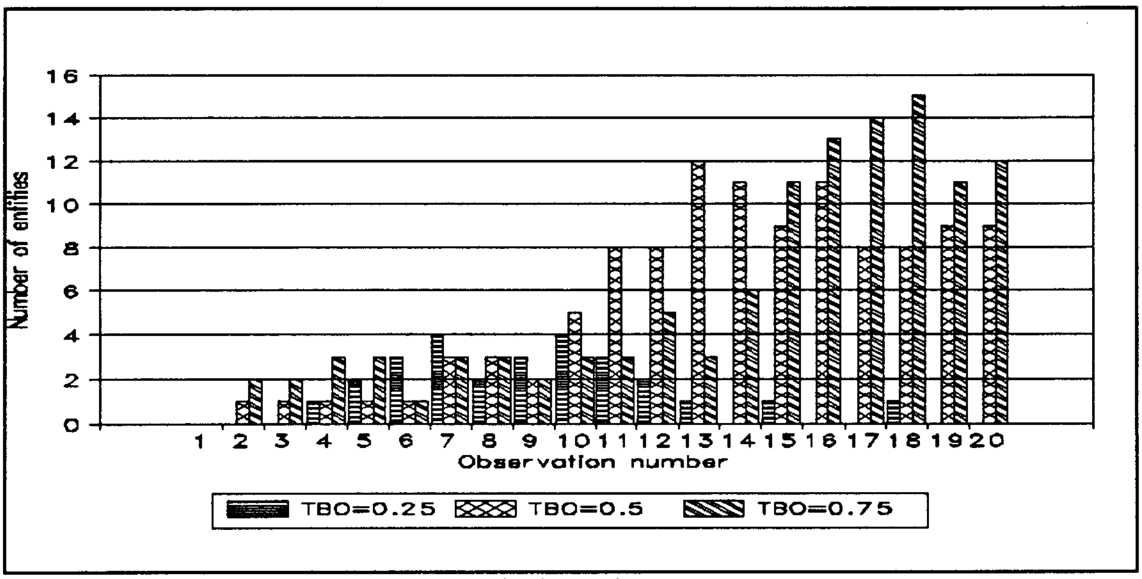


Figure 2.1 : Number of entities in the system for a two-machine system, {TBO = 0.25; 0.5; 0.75}.

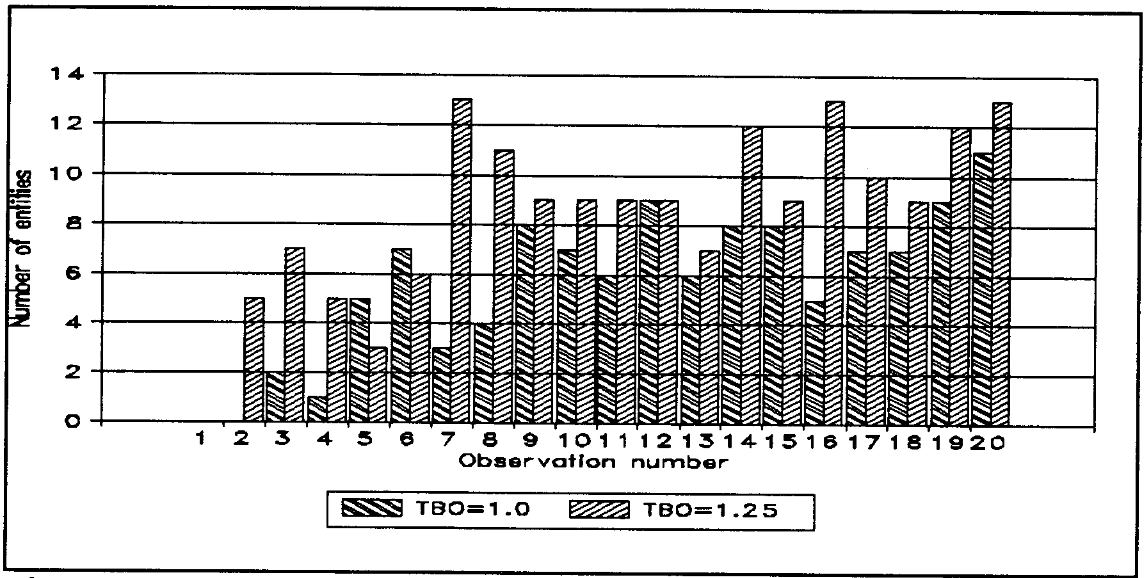


Figure 2.2 : Number of entities in the system for a two-machine system, {TBO = 1.0; 1.25}.

Similarly, there are two observations of an empty system for TBO of 1.0 and one observation for TBO of 1.25, as shown in Figure 2.2. Since the system started empty and idle, there is at least one observation with system empty.

An empirical distribution of initial observations of an empty, idle system is established by including all 1000 replications for a given TBO. A summary of the mean occurance and standard deviation of initial observations (using equations 1.33 and 1.18, respectively) as well as the maximum occurance of initial observations from a particular replicate is listed in Table 2.8.

For TBO of 0.25 time units, for example, the mean occurance is 3.4470 and at most there are 16 initial observations of empty, idle system recorded from a particular replication. Table 2.8 shows that the mean occurance and standard deviation decrease as the TBO increases.

Table 2.8 : Empirical distributions of number of initial observations of empty, idle system.

TBO	Mean	Std. dev.	Max
0.25	3.4470	0.0815	16
0.5	2.5440	0.0544	12
0.75	2.1280	0.0421	9
1.0	1.9830	0.0387	8
1.25	1.8000	0.0324	6

The ANOVA results (Table 2.9) show that the main effect of parameter time between observations (TBO) is significant, while the main effect of parameter batch size and their interaction are not significant.

Table 2.9 : ANOVA for a two-machine system with initial observations of empty, idle system as the dependent variable.

Dependent variable : Initial observations					
Source	DF	Anova SS	Mean Square	F value	Pr > F
TBO	4	3445.7944	861.4486	309.35	0.0001
Batch	1	0	0	0	1.0000
TBO*Batch	4	0	0	0	1.0000

The DMRT on mean occurance of initial observation shows that empirical distribution means are significantly different from each other for the different values of TBO used in the study (Table 2.10).

Table 2.10 : Ranked mean number of initial observations for a two-machine system.

Duncan Grouping	Mean	N	COND
A	3.4470	2000	0.25
B	2.5440	2000	0.5
C	2.1280	2000	0.75
D	1.9830	2000	1.0
E	1.8000	2000	1.25

Means with the same letter are not different significantly.

2.4.4 Empirical truncation point distribution.

For every design level, an empirical truncation point distribution is established by including all independent truncation points recorded from 1000 independent replications. The frequency distributions of the truncation point for all designs are given in Figures 2.3 and 2.4.

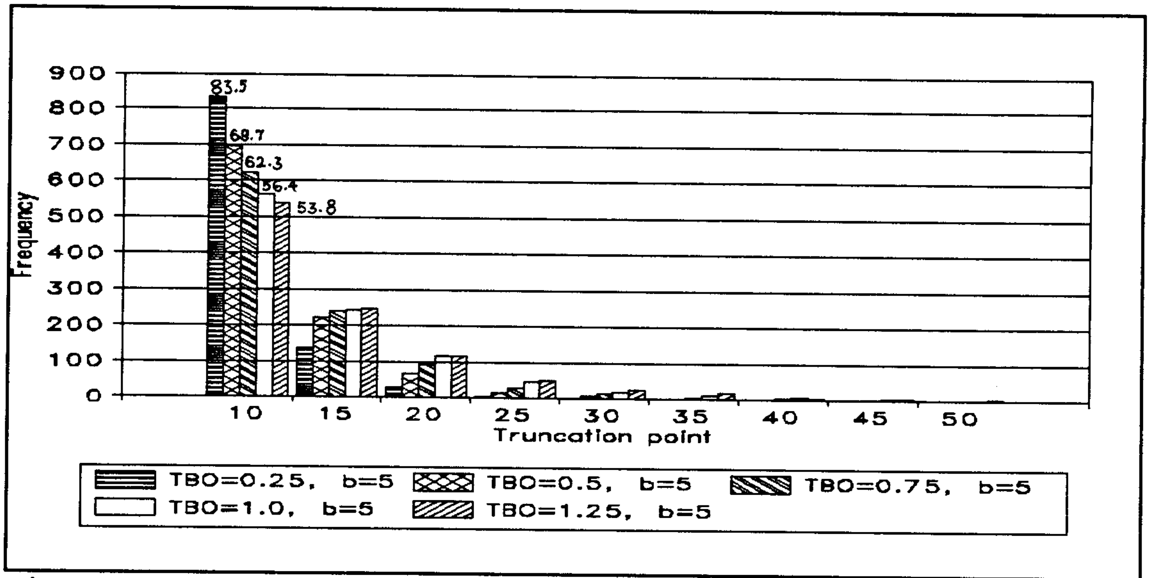


Figure 2.3 : Frequency distribution of the truncation point with b = 5.

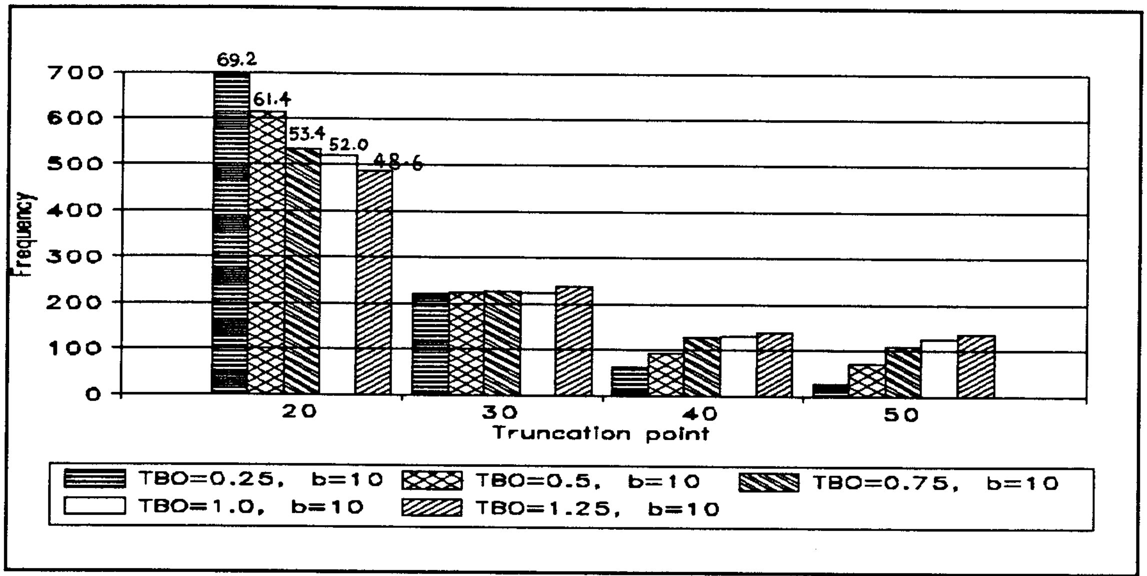


Figure 2.4 : Frequency distribution of the truncation point with b = 10.

A summary of the mean of truncation point ($\bar{d}$, using equation 1.35), standard deviation (using equation 1.18), and the mode is listed in Table 2.11.

Table 2.11 : Empirical truncation point distributions for a two-machine system.

Design level (TBO, b)	Mean	Std. dev.	Mode
0.25, 5	10.98	0.0748	10
0.25, 10	24.21	0.2285	20
0.5, 5	12.05	0.1136	10
0.5, 10	26.18	0.2900	20
0.75, 5	12.945	0.1507	10
0.75, 10	28.13	0.3264	20
1.0, 5	13.865	0.1838	10
1.0, 10	28.60	0.3364	20
1.25, 5	14.3150	0.1999	10
1.25, 10	29.27	0.3419	20

The frequency distributions and the table show that :

- The frequency of truncation point decreases as the truncation point increases for any design level.
- For batch count k = 2, the minimum and mode of truncation points for b = 5 and b = 10 are d = 10 and d = 20, respectively. These minimum truncation points satisfy the minimum requirement of d = k x b.
- As the TBO increases for a batch size of five, the frequency of d = 10 decreases while the frequency for other truncation points greater than d increases. This condition also occurs for a batch size of 10 with d = 20.
- As the batch size increases for a given TBO, the minimum and standard deviation of truncation points increase. Also notice, as expected, the mean truncation point increases.
- As the TBO increases for a given batch size, the mean and standard deviation of truncation point increase.

The ANOVA results (Table 2.12) show that the main effect of parameters time between observations and batch size as well as their interaction are significant.

Finally, DMRT on the means of truncation point (Table 2.13) shows that there is no significant difference in means for TBO = 1.0 and TBO = 0.75 time units with a batch size of 10 as well as in means for TBO = 1.0 and TBO = 1.25 time units with a batch size of five, while there is significant difference in means for other combinations of TBO's and batch sizes.

Table 2.12 : ANOVA for a two-machine system with empirical truncation point as the dependent variable.

Dependent variable : Truncation point					
Source	DF	Anova SS	Mean Square	F value	Pr > F
TBO	4	23146.835	5786.7087	98.44	0.0001
Batch	1	521789.52	521789.52	8876.04	0.0
TBO*Batch	4	12363.615	308.4038	5.25	0.0003

Table 2.13 : Ranked truncation point means for a two-machine system with $\alpha = 10\%$.

Duncan Grouping	Mean	N	COND
A	29.270	1000	1.25, 10
B	28.600	1000	1.0, 10
B	28.130	1000	0.75, 10
C	26.180	1000	0.5, 10
D	24.210	1000	0.25, 10
E	14.315	1000	1.25, 5
E	13.865	1000	1.0, 5
F	12.945	1000	0.75, 5
G	12.050	1000	0.5, 5
H	10.980	1000	0.25, 5
Means with the same letter are not different significantly.			

2.5 <u>Interpretation of Results</u>

The effects of parameter selection on performance measures - the steady-state mean, the average mean-squared error, the initial observations of empty, idle system, and the empirical truncation point distribution - will be discussed in this section.

2.5.1 Interpretation for the steady-state means.

The steady-state mean number of entities in the system increases as the TBO increases because increasing the TBO will significantly reduce the number of initial observations of an empty, idle system (i.e. the zero-entity observations) recorded from the steady-state period. Hence any single mean calculated based on the observations having less zero-entity observations will be greater than the single mean calculated based on the observations having more zero-entity observations.

Increasing the batch size will increase the truncation point meaning more observations are discarded. This reduces the number of observations recorded from the transient period used in estimating the steady-state mean. From time series analysis it is known that including more observations will dampen or smooth the average value. Thus, increasing the truncation point or discarding more observations will give a higher average result than otherwise.

However, it is important to note that discarding more observations by increasing the batch size is not always desirable since the mean-squared error of overall mean increases significantly. Smaller mean-squared error may be achieved by using small batch sizes.

2.5.2 Interpretation for the average mean-squared error.

The average mean-squared error increases as the truncation point increases, that is, as more observations are discarded. Even though some independent runs from any design level show that the bias decreases and the variance increases as the truncation point increases, the average bias and average variance from 1000 replications show the contrary. The only cause for this variability as well as the variability of the average mean-squared error is the sequence of random number selected (automatically) for each replication (see Baxter [1990]). Although using multiple replications to minimize the extreme results still shows the unexpected results, it is recommended to still use the multiple replications and to estimate the mean-squared error by adding the bias and variance of the steady-state mean instead of by averaging the mean-squared errors recorded from run to run for each design level.

Baxter's results for a single-machine system using the Schriber's rule support the results of this study. Theoretically, these two results show that the average variance increases as more observations are discarded for a fixed sample size. The results of this study and Baxter's study [1990] are contrary to Wilson's [1977, 1978a, and 1978b] results for a one-machine system due to the experimental nature of this study as compared to the experimental nature used by Wilson. A given fixed range of truncation points and set of initial conditions were used in Wilson's study, while in this study, the truncation points were determined experimentally by applying Schriber's truncation rule.

2.5.3 Interpretation for the number of initial observations.

As mentioned earlier, increasing TBO will significantly reduce the number of initial observations. This is because for a longer TBO there will be a greater chance of one of several activities (such as arrival, waiting, in service, blocking, and departure) occurring at the time the observations are recorded. Since these initial observations could be recorded from the transient and steady-state periods, decreasing the number of initial observations by increasing the TBO would also minimize the chance of recording the initial observations (the zero-entity observations) during the steady state period.

The variance of the number of initial observations also decreases for a larger TBO since the number of entities in the system will become more stable as the simulation time elapses. Therefore, it is recommended not to use a small TBO (such as 0.25 time units).

2.5.4 Interpretation for the empirical truncation point distributions.

The frequency distributions of the minimum truncation points imply that for a small TBO with a batch size of five, most of the truncation is done at d = 10; that is, discarding the first 10 observations in order to estimate the steady-state mean. Also, with small TBOs, it has been shown that there is a significant number of initial observations of an empty, idle system (Figure 2.1). The combined effect can be an incorrect indication that the steady-state condition has been achieved. Thus, increasing the TBO will not only reduce the number of initial observations of empty, idle system and the chance of including these zero-entity observations in estimating the steady-state mean, but the premature truncation process can also be avoided.

The minimum truncation points for both batch sizes satisfy the minimum requirement of truncation point $d = k \times b$, where k is the batch count (k = 2 batches in this study). In pairwise comparison the earliest truncation process can happen at the $2b^{th}$ observation. As the batch size increases from five to 10 observations per batch, the mean of truncation point distribution increases since its minimum truncation point also increases.

2.6 Sensitivity Analysis of Results to Queue Size

In order to analyze the sensitivity of results to queue size, a system which consists of two M/M/1/ ∞ models - each with infinite queue capacity - is selected.

Due to the infinite queue capacity, no customer will block and balk from the system. As expected, the theoretical steady-state mean number of entities in the system (μ_x) will increase from 10.2631 to 18 entities. Besides, it is reasonable to assume that longer time is required to reach the steady-state conditions due to the increase in the number of entities in the system.

The results with infinite queue capacity show similarity to the two-machine system with finite queue capacity evaluated earlier; increasing the TBO will reduce the number of initial observations and discarding more observations by increasing the batch size will improve the bias but will worsen the mean-squared error.

However, with infinite queue size, a longer TBO should be used because time to reach the steady-state conditions with infinite queue size is longer than with finite queue size. It is apparent that the values selected for parameters TBO and batch size are a function of input/output rates, that is system-dependent. Thus, for a two-machine system with infinite queue size, it is suggested to use a TBO of five time units with a batch size of five.

2.7 Summary of Results

- Use multiple replications to minimize the extreme results in determining the steady-state mean and the truncation point distributions.
- Calculate the mean-squared error by adding the bias and variance of overall steady-state mean instead of by averaging the mean-squared errors recorded from 1000 runs.
- Use a batch size of five since discarding more observations by increasing the batch size from five to 10 observations per batch will inevitably increase the mean-squared error of overall steady-state mean.
- Use TBO greater than 0.25 time units because increasing the TBO will reduce the number of initial observations so that the chance of recording these observations to estimate the steady-state mean can be minimized. With a batch size of five, TBO = 0.5 time units, followed by TBO = 0.75 time units, will minimize the mean-squared error and, hence, the selection of a batch size of five with TBO = 0.5 time units (or with TBO = 0.75 time units) should be used for a two-machine system.

A summary of confidence intervals (c.i.) with different α's for the recommended design levels is presented in Table 2.14.

Table 2.14 : Confidence Intervals with diferent α's for a two-machine system, b = 5.

TBO	Mean	C.I.		
		α = 0.01	α = 0.05	α = 0.1
.5	9.953	9.420;10.485	9.635;10.270	9.686;10.219
.75	11.655	11.039;12.272	11.288;12.023	11.347;11.964

CHAPTER 3. APPLICATION TO A THREE-MACHINE SYSTEM

3.1 Analytical Model

The second queueing system evaluated in this study is a system consisting of three M/M/1/15 models known as three-machine system with three queues placed in tandem. The arrival rate λ and the service rate μ for each machine are still the same as used in the previous system, that is 4.5 arrivals per time unit and five jobs per time unit, and are distributed exponentially with a finite queue capacity of 15 per queue. The utilizations of the first, second, and third machines are $\rho_1 = 0.9$, $\rho_2 = 0.877$, and $\rho_3 = 0.860$, respectively. The theoretical steady-state mean number of entities in the system μ_x is 14.8333. See Appendix III for more details about these analytical results.

3.2 Computer Model

The computer programming was accomplished by using SIMAN and FORTRAN languages and the only new subroutine added to the program developed for the two-machine case was subroutine M2TOM3 though there were some changes in subroutines M1TOM2 and ENDOS. Other subroutines used in the previous program for a two-machine system are also used in this program without making any changes. The flowcharts for subroutines M2TOM3 and ENDOS are given in Appendices IV-A and IV-B, respectively.

3.2.1 Subroutine M1TOM2 (start processing on machine-2).

The functions of this subroutine are as follows :

- to process the job sent from machine-1 and schedule its completion time on machine-2 if machine-2 is idle. After the process is completed, the job is sent to machine-3. In a two-machine system, the job is sent to leave the system after being processed on machine-2.
- to make the job wait in the second queue if machine-2 is busy and a space is available in that queue.
- to block machine-1 if machine-2 is busy and no space is available in the second queue. If blockage occurs, machine-1 will not process a new job while machine-2 keeps processing the current job.
- to process the waiting job in the first queue (if any) every time machine-1 is idle and, then, schedule its completion time.

3.2.2 Subroutine M2TOM3 (start processing on machine-3).

The functions of this subroutine are as follows :

- to process the job sent from machine-2 and schedule its completion time on machine-3 if machine-3 is idle. After the process is completed, the job is sent to leave the system.
- to make the job wait in the third queue if machine-3 is busy and a space is available in that queue.
- to block machine-2 if machine-3 is busy and no space is

available in the third queue. If the blockage occurs, machine-2 will not process a new job while machine-3 keeps processing the current job.

- to unblock machine-1 whenever the job is sent from machine-2 to machine-3 so that there is an available space in the second queue. Then, the blocking job is moved from machine-1 to the second queue.
- to process the waiting job in the first queue (if any) every time machine-1 is idle and, then, schedule its completion time.

3.2.3 Subroutine ENDOS (leave the system).

The functions of this subroutine are as follows :

- to process the waiting job in the third queue (if any) every time the job is sent to leave the system.
- to process the waiting job in the second queue (if any) every time machine-2 is idle and, then, schedule its completion time to leave machine-2.
- to unblock machine-2 whenever there is an available space in the third queue by moving the blocking job from machine-2 to that queue.
- to unblock machine-1 whenever there is an available space in the second queue by moving the blocking job from machine-1 to that queue.
- to process the waiting job in the first queue (if any) every time machine-1 is unblocked and idle. Then schedule

its completion on machine-1 and send it to machine-2.

3.3 Experimental Design

It is reasonable to assume that more machines in the system, more time will be spent by a job in the system, so that there will be a smaller chance that the system is empty at the time the observations are performed. Hence, shorter TBO's can be used in the three-machine system. The experimental design for a three-machine system is a 3 x 2 design with three and two levels for parameters time between observations and batch size, respectively, as shown in Table 3.1. Each design level is analyzed based on 1000 runs.

Table 3.1 : Parameters and factor levels (3 x 2) for a three-machine system's experimental design.

Parameters	Factor levels
T : time scale [average arrivals / observations]	1.6875; 2.25; 2.8125
TBO : time between observations [time units]	0.375; 0.5; 0.625
b : batch size [number of observations per batch]	5; 10

3.4 Results

Again, equations 1.16, 1.17, 1.18, 1.19, and 1.21 are used to calculate the grand mean, sample variance, standard error (standard deviation), bias, and mean-squared error, respectively.

A summary of the steady-state mean number of entities, bias, variance, mean-squared error, and standard error for a three-machine system is presented in Table 3.2. The table shows that the variance, mean-squared error, and standard error are minimized with the parameter set of TBO = 0.375 time units and a batch size of five. However, the bias is minimized with the parameter set of TBO = 0.625 time units and a batch size of five.

Table 3.2 : Overall steady-state results for a three-machine system, with an α of 10%.

TBO, b	Mean	Bias	Var.	MSE	s.e.
0.375, 5	12.4615	-2.3718	31.0541	36.6795	0.1762
0.375, 10	13.7861	-1.0472	49.3181	50.4146	0.2221
0.5, 5	14.0700	-0.7633	37.2119	37.7945	0.1929
0.5, 10	15.4563	0.6230	56.1407	56.5288	0.2369
0.625, 5	15.4544	0.6211	43.4883	43.8741	0.2085
0.625, 10	17.0085	2.1752	67.7359	72.4674	0.2603

Note :
- λ = 4.5 arrivals per time unit.
- TBO : time between observations [time units].
- There are 1000 replications used in each design level to construct the single-point estimate, mean number of entities in the system.
- Bias represents the deviation between the overall mean and the theoretical steady-state number of entities (μ_x = 14.8333), while variance represents the variability within the overall mean. Finally, MSE is the summation of squared bias and variance.

3.4.1 Steady-state mean number of entities in the system.

The two-way ANOVA test and DMRT are then performed on the steady-state mean number of entities and the results are presented in Tables 3.3 and 3.4.

Table 3.3 : ANOVA for a three-machine system with steady-state mean as the dependent variable.

Dependent variable : Mean					
Source	DF	Anova SS	Mean Square	F value	Pr > F
TBO	2	9667.057	4833.529	101.78	0.0001
Batch	1	3031.815	3031.815	63.84	0.0001
TBO*Batch	2	14.101	7.051	0.15	0.8620

These tables show that :

- The main effects for the two parameters TBO and batch size are significant, while the interaction between these parameters is not significant.

Table 3.4 : Ranked steady-state means by using DMRT with α = 10%.

Duncan Grouping	Mean	N	COND
A	17.009	1000	0.625, 10
B	15.456	1000	0.5, 10
B	15.454	1000	0.625, 5
C	14.070	1000	0.5, 5
C	13.786	1000	0.375, 10
D	12.462	1000	0.375, 5
Means with the same letter are not different significantly.			

- As the TBO increases from 0.375 to 0.625 time units for a fixed batch size, the steady-state mean increases.
- As the batch size increases for a given TBO, the steady-

state mean increases.

- There is no significant difference in means for parameter set of TBO = 0.375 with b = 10 and parameter set of TBO = 0.5 with b = 5 as well as in means for parameter set of TBO = 0.5 with b = 10 and parameter set of TBO = 0.625 with b = 5. However, the means for other parameter sets are significantly different from each other.

It is evident that these results are similar to the two-machine system evaluated earlier; increasing the TBO will reduce the number of initial observations (the zero-entity observations) recorded from the steady-state period so that the chances of a single mean with less zero-entity observations will be greater than that of with more zero-entity observations. Likewise, discarding more observations by increasing the batch size improves the bias but worsens the mean-squared error.

3.4.2 Average mean-squared error ($\overline{MSE}$).

A summary of the average squared bias, average variance, and average mean-squared error, which were estimated from run to run using equations 1.31, 1.30, and 1.29, respectively, are presented in the following Table 3.5. The two-way ANOVA test and DMRT are then performed on the average mean-squared error and the results are presented in the Tables 3.6 and 3.7.

Table 3.5 : Average results for a three-machine system with α = 10%.

TBO	Batch size b	Mean	$\overline{Bias^2}$	$\overline{Var}$	$\overline{MSE}$
0.375	5	12.4615	34.8561	20.7139	w.5700
0.375	10	13.7861	58.6806	11.9980	70.6786
0.5	5	14.0700	49.6672	25.9181	75.5853
0.5	10	15.4563	78.0541	15.5474	93.6015
0.625	5	15.4544	68.3941	31.2852	99.6793
0.625	10	17.0085	103.1684	18.7178	121.8862

Table 3.6 : ANOVA for a three-machine system with $\overline{MSE}$ as the dependent variable.

Dependent variable : $\overline{MSE}$					
Source	DF	Anova SS	Mean Square	F value	Pr > F
TBO	2	2181633.299	1090816.649	70.86	0.0001
Batch	1	510268.716	510268.716	33.15	0.0001
TBO*Batch	2	12733.595	6366.797	0.41	0.6631

These results show that :

- The main effects for the two parameters TBO and batch size are significant.
- As the batch size increases for a fixed TBO, the average squared bias and the average mean-squared error increase while the average variance decreases.
- As the TBO increases for a fixed batch size, the average squared bias, average variance, and average mean-squared error increase.

Table 3.7 : Ranked $\overline{MSE}$'s by using DMRT with α = 10%.

Duncan Grouping	$\overline{MSE}$	N	COND
A	120.886	1000	0.625, 10
B	98.679	1000	0.625, 5
B	93.601	1000	0.5, 10
C	75.585	1000	0.5, 5
C	70.679	1000	0.375, 10
D	55.570	1000	0.375, 5

$\overline{MSE}$'s with the same letter are not significantly different.

- There is no significant difference in average mean-squared errors for parameter set of TBO = 0.375 with b = 10 and parameter set of TBO = 0.5 with b = 5 as well as in average mean-squared errors for parameter set of TBO = 0.5 with b = 10 and parameter set of TBO = 0.625 with b = 5.
- The average mean-squared error is minimized with the parameter set of TBO = 0.375 time units with a batch size of five.

These results are similar to the results of a two-machine system, where the average variance decreases and the average squared bias increases as the truncation point increases. Since theoretically the average squared bias decreases and the average variance increases as more observations are discarded for a fixed sample size, it is recommended not to use the average mean-squared error (equation 1.29) but to use the overall mean-squared error (equation 1.21 as the summation of the bias and variance of the overall steady-state mean) to estimate the mean-squared error.

3.4.3 Initial observations of empty, idle system.

Figure 3.1 shows the first twenty observations taken from the first replication for three different values of TBO. From these observations there are at least 7, 1, and 1 observations of an empty, idle system for TBO = 0.375, 0.5, and 0.625 time units, respectively.

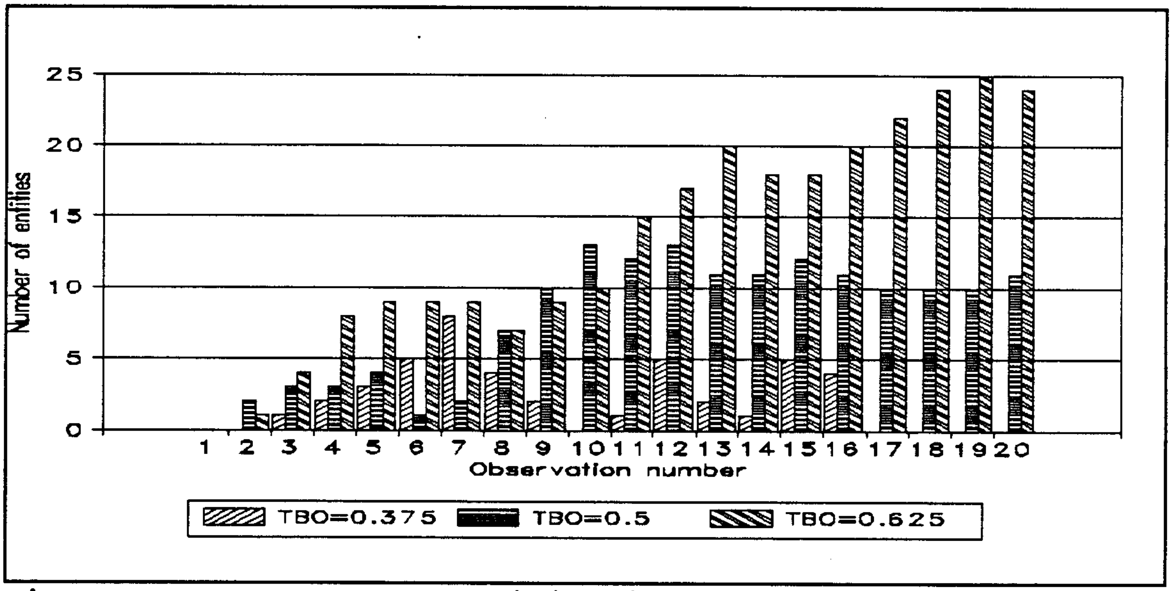


Figure 3.1 : Number of entities in the system for a three-machine system, {TBO = 0.375; 0.5; 0.625}.

A summary of the mean occurance (equation 1.33), standard deviation (equation 1.18), and the maximum occurance of the initial observations of an empty, idle system of all design levels is presented in Table 3.8.

Table 3.8 : Empirical distributions of number of initial-observations of empty, idle system.

TBO	Mean	Std. dev.	Max.
0.375	1.703	0.0333	7
0.5	1.396	0.0238	6
0.625	1.375	0.0210	5

The ANOVA results (Table 3.9) indicate that only main effect of parameter time between observations (TBO) is significant on the dependent variable - initial observations of empty, idle system - while the main effect of parameter batch size and their interaction are not significant.

Table 3.9 : ANOVA for a three-machine system with initial observations of empty, idle system as the dependent variable.

Dependent variable : Initial observations					
Source	DF	Anova SS	Mean Square	F value	Pr > F
TBO	2	134.8493333	67.4246667	95.75	0.0001
Batch	1	0	0	0	1.0000
TBO*Batch	2	0	0	0	1.0000

Furthermore, the results from DMRT on mean occurance of initial observations are given in Table 3.10. The results show that the empirical distributions means with TBO = 0.5 and TBO = 0.625 time units are not significantly different; additionally, increasing TBO will decrease the mean occurance and standard deviation of initial observations.

Table 3.10 : Ranked mean number of initial observations for a three-machine system.

Duncan Grouping	Mean	N	COND
A	1.703	2000	0.375
B	1.396	2000	0.5
B	1.375	2000	0.625

Means with the same letter are not different significantly.

These results are similar to the two-machine system. Since the chance of recording the initial observations from the steady-state period is minimized for a longer TBO, it is recommended to use TBO = 0.375 or TBO = 0.5 time units.

3.4.4 Empirical truncation point distribution.

The summary of the mean truncation point (equation 1.35) standard deviation (equation 1.18), and the mode is shown in Table 3.11.

Table 3.11 : Empirical truncation point distributions for a three-machine system.

Design level (TBO, b)	Mean	Std. dev.	Mode
0.375, 5	12.7300	0.1308	10
0.375, 10	28.9200	0.3123	20
0.5, 5	13.3550	0.1401	10
0.5, 10	29.3200	0.3250	20
0.625, 5	14.1800	0.1625	10
0.625, 10	31.0900	0.3365	20

The frequency distributions of the truncation points for all designs are depicted in Figures 3.2 and 3.3.

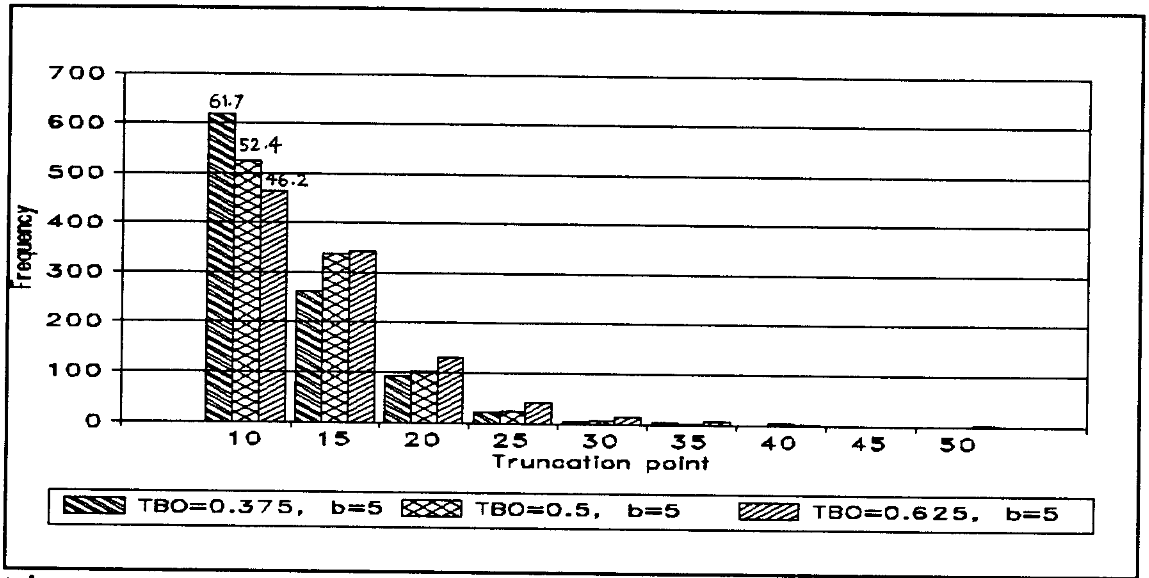


Figure 3.2 : Frequency distribution of the truncation point with b = 5.

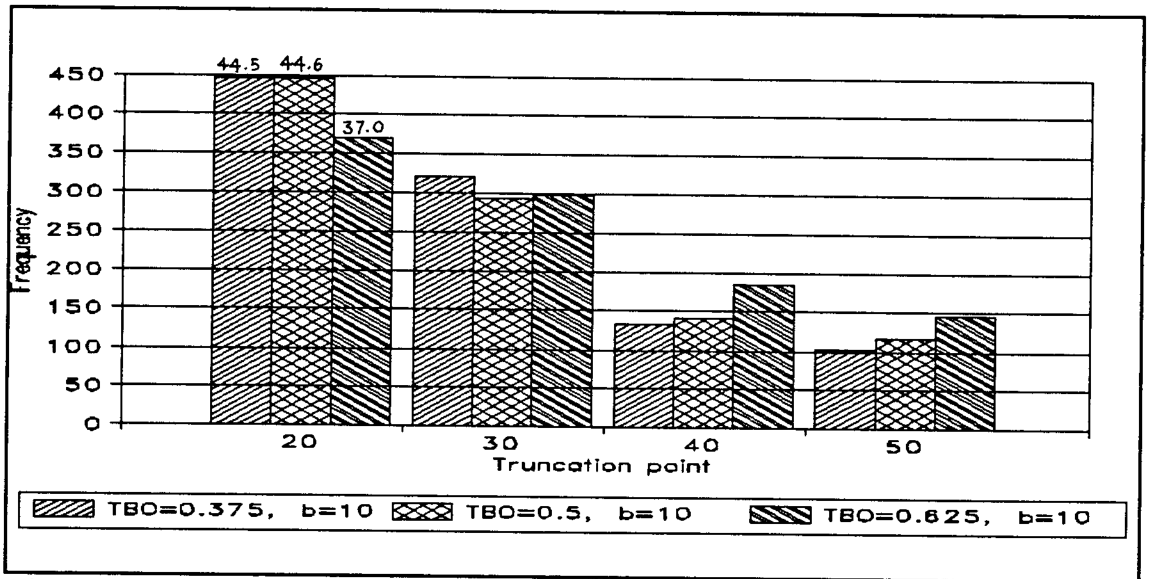


Figure 3.3 : Frequency dsitribution of the truncation point with b = 10.

The ANOVA results (Table 3.12) indicate that the main effects of parameters time between observations (TBO) and batch size (b) are significant on the dependent variable, empirical truncation point.

Table 3.12 : ANOVA for a three-machine system with empirical truncation point as the dependent variable.

Dependent variable : Truncation point					
Source	DF	Anova SS	Mean Square	F value	Pr > F
TBO	2	3481.5083	1740.7542	27.52	0.0001
Batch	1	401229.0375	401229.0375	6342.6	0.0
TBO*Batch	2	243.6750	121.6750	1.93	0.1458

Performing DMRT on the means of the truncation point proved that there is no significant difference in means for TBO = 0.5 and TBO = 0.625 time units with a batch size of five, while there is significant difference in means for other design levels (Table 3.13).

Table 3.13 : Ranked truncation point means for a three-machine system with α = 10%.

Duncan Grouping	Mean	N	COND
A	31.090	1000	0.625, 10
B	29.320	1000	0.5, 10
B	28.920	1000	0.375, 10
C	14.180	1000	0.625, 5
D	13.355	1000	0.5, 5
E	12.730	1000	0.375, 5
Means with the same letter are not different significantly.			

Again, the effects of changing the TBO and batch size independently on the truncation point distributions for a three-machine system are similar to the two-machine system in such a way that increasing the TBO will reduce the number of initial observations and prevent the premature truncation process (shown by the decrease in the frequency of the minimum truncation points). In addition, discarding more observations by increasing the batch size will increase the minimum truncation point and, obviously, also the mean truncation point.

It is evident that the general recommendations for a two-machine system can also be used for a three-machine system though specific parameter values may be different. For a three-machine system, it is recommended to use TBO = 0.375 time units because with a batch size of five the mean-squared error for TBO = 0.375 is less than that for TBO = 0.5 time units. Increasing the number of machines in the system from two to three machines requires a reduction in the value of TBO; as more machines (processes) are required to finish a job or an entity, more time is spent in the system and more jobs will stay longer in the system. Thus through smaller TBO there will be a smaller chance of recording the initial observations of empty, idle system (the zero-entity observations).

A summary of the confidence intervals with different α's for the recommended design levels is presented in Table 3.14 below.

Table 3.14 : Confidence Intervals with diferent α's for a three-machine system, b = 5.

TBO	Mean	C.I.		
		α = 0.01	α = 0.05	α = 0.10
.375	12.46	11.881;13.041	12.116;12.809	12.172;12.751
.5	13.79	13.055;14.517	13.351;14.221	13.421;14.151

CHAPTER 4. CONCLUSIONS, IMPLICATIONS, AND FUTURE RESEARCH

4.1 Conclusions

In order to study the behavior of Schriber's truncation rule used for a complex queueing system with machine in series (or in tandem), two different models - a two-machine system and a three-machine system - were selected in this study. The arrival rate of 4.5 arrivals per time unit and service rate of five jobs or entities per time unit were distributed exponentially.

The point estimates or performance measures evaluated to describe the effects of changing parameters time between observations (TBO) and batch size (b) on the application of Schriber's truncation rule are as follows :

- Mean number of entities in the system.
- Average mean-squared error, $\overline{MSE}$.
- Number of initial observations with empty, idle system.
- Truncation point.

Although it has been stated in literature that the best initial or startup condition is the steady-state mode or close to steady-state mode, the empty, idle system was chosen in this study because this condition is the most convenient and more likely to be used in practice. Furthermore, deleting some biased observations existed during the transient period is found to be an effective, efficient technique for reducing the effect of initialization bias induced by any artificial startup condition.

As mentioned earlier, the parameters varied were time between observations (TBO) and batch size (b), while batch count (k) and tolerance (ϵ) were held constant. Because of one of the Markovian property in queueing theory, time between observations does not have an effect on the transient time. However, time between observations in simulation studies has an effect on reaching the steady-state conditions.

For a two-machine system, the ANOVA results indicate that the main effects of parameters time between observations (TBO) and batch size (b) are significant for the dependent variable steady-state mean. The average mean-squared error and empirical truncation point were affected by the paremeters time between observations, batch size, and their interaction. Besides, the number of initial observations with empty, idle system was affected only by time between observations.

For a three-machine system, the ANOVA results show that the steady-state mean, average mean-squared error, and empirical truncation point were affected by both independent parameters - time between observations and batch size, while the number of initial observations with empty, idle system was affected only by time between observations.

A bacth size of five is recommended to be used in simulation studies for the system evaluated because increasing the batch size from five to 10 will delete more observations and will inevitably worsen the mean-squared error of the overall steady-state mean. Thus, it is apparent that using a batch size of five will prevent an excessive truncation and will detect the occurance of the steady-state operating condition more rapidly.

The time between observations should be selected to be short enough so that the chances of recording the initial observation with empty, idle system indicating the extreme initialization bias can be minimized resulting in more consistent steady-state mean. For the two-machine system with a batch size of five, TBO = 0.5 time units, followed by TBO = 0.75 time units, minimized the mean-squared error of the overall steady-state mean. For the three-machine system, TBO = 0.375 time units with a batch size of five did the same. Figure 4.1 shows the suggested values of parameter TBO for different number of machines placed in series in the system.

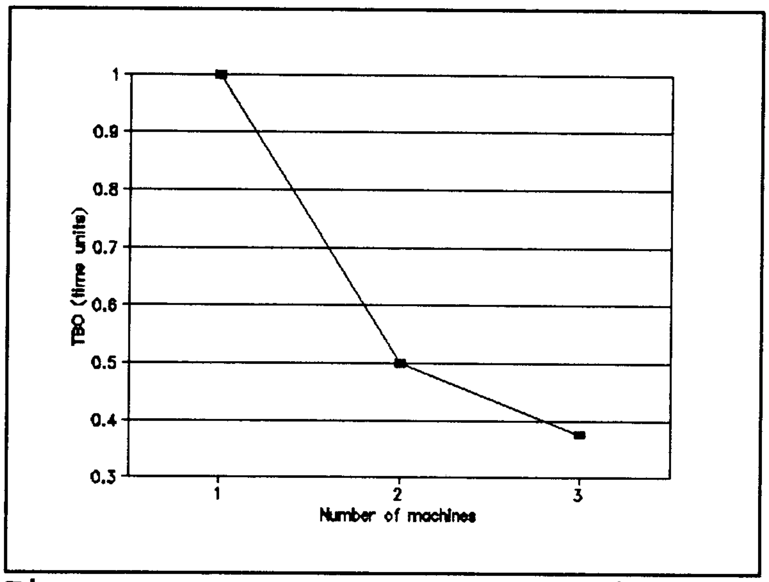


Figure 4.1 : Suggested TBO for different number of machines.

It is evident that with more machines placed in series in a system, a smaller TBO can be used. This is because more time will be spent in the system to finish a job; consequently jobs will stay longer in the system so that chances of recording the initialization bias is minimized with smaller TBO.

Due to the randomness caused by the use of certain sequences of random numbers, it is recommended to use multiple replications to minimize the extreme results in determining the overall steady-state mean, number of initial observations with empty, idle system, and truncation point. It is apparent that these measures have probability distributions associated with them. Furthermore, because of this randomness, it is also suggested not to estimate the mean-squared error by averaging the mean-squared errors recorded from 1000 runs. The best way to estimate the mean-squared error is by adding the squared-bias and variance of the overall steady-state mean.

4.2 Implications

Identifying truncation point for steady-state results has an important applications in system simulation, particularly for complex, large-scale system with high degree of variability. The results from this research (together with the initials results from Baxter [1990]) can be extended to complex systems that consist of stages in series. These are not just manufacturing systems, but also include such application areas as reliability studies of components in series. To iterate, the critical parameters in these systems are the batch size and time between observations; the exact values selected for these parameters are a function of input/ouput rates, but the relationships established in this study can be generalized to other situations.

4.3 Future Research

This research can be extended in a number of situations. Two of these are :

- Evaluation of hybrid systems, where the system consists of machines (stages) in series and in parallel.
- Comparing the performance of Schriber's rule with the performance of some of the other rules reported in literature (Gafarian et al [1978], Kelton and law [1984], and Schruben [1982]).

BIBLIOGRAPHY

Banks, J. and J.S. Carson, "Discrete-Event System Simulation", Prentice-Hall Inc., Englewood Cliffs, NJ, 1984.

Baxter, L., "Truncation Rules in Simulation Analysis : Effect of Batch Size, Time Scale, and Input Distribution on the Application of Schriber's Rule", MS Thesis, Dept. of Industrial Engineering Oregon State University, 1990.

Cody, Ronald P. and Jeffrey K. Smith, "Applied Statistics and the SAS Programming Language", 2nd edition, North-Holland, New York, 1987.

Crane, M.A. and D.L. Iglehart, "Simulating Stable Stochastic Systems : I. General Multiserver Queues", J. Assoc. Comput. Mach. 21, 103-113, 1974.

Crane, M.A. and D.L. Iglehart, "Simulating Stable Stochastic Systems : III. Regenerative Processes and Discrete Event Simulations", Operations Research v23, 33-45, 1975.

Fishman, G.S., "Estimating Sample Size in Computer Simulation Experiment", Management Science v18, 21-38, 1971.

Fishman, G.S., "Bias Considerations in Simulation Experiments", Operations Research v20, 685-790, 1972.

Fishman, G.S., "Concepts and Methods in Discrete-Event Digital Simulation", Wiley, New York, 1973.

Fishman, G.S., "Principles of Discrete Event Simulation", Wiley, New York, 1978.

Gafarian, A.V., C.J. Ancker, Jr., T. Morisaku, "Evaluation of Commonly Used Rules for Detecting Steady State in Computer Simulation", Naval Research Logistics Quarterly v25, 511-529, 1978.

Kelton, W.D. and A.M. Law, "Simulation Modeling and Analysis", McGraw-Hill Inc., 1982.

Kelton, W.D. and A.M. Law, "A New Approach for Dealing with the Startup Problem in Discrete Event Simulation", Naval Research Logistics Quarterly v30, 641-658, 1983.

Kelton, W.D. and A.M. Law, "An Analytical Evaluation of Alternative Strategies in Steady State Simulation", Operations Research v32, 1, 169-184, 1984.

Law, A.M., "Confidence Intervals in Discrete Event Simulation

: A Comparison of Replication Means and Batch Means", Naval Research Logistics Quarterly v24, 667-678, 1977.

Law, A.M. and J.S. Carson, "A Sequential Procedure for Determining the Length of a Steady State Simulation", Operations Research v27, 5, 1011-1025, 1979.

Law, A.M., "A Tutorial on Statistical Analysis of Simulation Output Data", Proceedings the 1980 Winter Simulation Conference.

Law, A.M., "Statistical Analysis of Simulation Output Data", Operations Research v31, 6, 983-1029, 1983.

Law, A.M. and W.D. Kelton, "Confidence Intervals for Steady State Simulations : I. A Survey of Fixed Sample Size Procedures", Technical Report 78-5, Dept. of Industrial Engineering, University of Wisconsin, 1979.

MacDougal, M.H., "Simulating Computer Systems : Techniques and Tools", The MIT Press, Cambridge, MA, 1978.

MacNair, E.A. and C.H. Sauer, "Elements of Practical Performance Modeling", Prentice-Hall Inc., Englewood Cliffs, NJ, 1985.

Neuts, M.F., "Probability", Allyn and Bacon, Boston, 1973.

Petersen, Roger G., "Design and Analysis of Experiments", Marcel Dekker Inc., New York, 1985.

Schmeiser, Bruce, "Batch Size Effects in the Analysis of Simulation Output", Operations Research v30, 3, 556-568,1982.

Schruben,L.W., "Detecting Initialization Bias in Simulation Output", Operations Research, v30, 569-590, 1982.

Schruben, L.W., H. Singh, and L. Tierney, "Optimal Tests for Initialization Bias in Simulation Output", Operations Research v31, 1167-1178, 1983.

Taha, Hamdy A., "Simulation Modeling and SIMNET", Prentice-Hall Inc., Englewood Cliffs, NJ, 1988.

Turnquist, M.A. and J.M. Sussman, "Toward Guidelines for Designing Experiments in Queueing Simulation", Simulation 28, 5, 137-144, 1977.

Wilson, J.R., "A Procedure for Evaluating Startup Policies in Simulation Experiments", MS Thesis, School of Industrial Engineering Purdue University, 1977.

Wilson, J.R. and A.A.B. Pritsker, "A Survey of Research on the Simulation Startup Problem", Simulation v31, 55-58, 1978.

Wilson, J.R. and A.A.B. Pritsker, "Evaluation of Startup Policies in Simulation Experiments", Simulation v31, 79-89, 1978.

Winston, Wayne L., "Operations Research : Applications and Algorithm", PWS-KENT Publishing Company, Boston, 1987.

APPENDICES

Appendix I : Theoretical Steady-State Mean Number of Entities in the Two-Machine System.

In order to calculate theoretical steady-state mean number of entities in the two-machine system (two machines in series or in tandem), the average number of entities calculated separately from each machine must be known first.

a. <u>The first machine (M/M/1/FIFO/15/ ∞).</u>

- The traffic intensity or load factor of the first machine with an arrival rate (λ) of 4.5 arrivals per time unit and a service rate (μ) of five entities per time unit is $\rho_1 = \lambda/\mu = 0.9$.
- The average number of entities in the i^{th} machine (L_i) is given by [Winston, 1987]

$$L_i = \frac{\rho_i[1 - (c+1)\rho_i^{c} + c\rho_i^{c+1}]}{(1 - \rho_i^{c+1})\ (1 - \rho_i)}$$

where : c = 15, queue capacitiy of the first machine.

Thus, the average number of entities in the first machine is

$$L_1 = \frac{0.9[1 - (15+1)0.9^{15} + 15 \times 0.9^{15+1}]}{(1 - 0.9^{15+1})\ (1 - 0.9)}$$

L_1 = 5.361 entities.

b. <u>The second machine (M/M/1/FIFO/15/ ∞).</u>

- The arrival rate of entities to the second machine depends on the arrival rate and service rate as well as on the queue capacity of the first machine. Thus, the effective rate of the second machine becomes

$$\bar{\lambda}_2 = \lambda(1 - P_{(15;1)})$$

where :

$P_{(15;1)}$: probability that an entity will balk from the first machine because no space is available in the first queue.

$$P_{(15;1)} = \rho_1{}^c \times \frac{1 - \rho_1}{1 - \rho_1{}^{c+1}}$$

$$P_{(15;1)} = 0.9^{15} \times \frac{1 - 0.9}{1 - 0.9^{15+1}}$$

$$P_{(15;1)} = 0.025.$$

Thus

$$\bar{\lambda}_2 = 4.5(1 - 0.025)$$

$\bar{\lambda}_2 = 4.386$ arrivals per time unit.

- The traffic intensity of the second machine with a service rate (μ) of five entities per time unit is

$$\rho_2 = \bar{\lambda}_2/\mu = 0.877.$$

The average number of entities in the second machine is

$$L_2 = \frac{0.877[1 - (15+1)0.877^{15} + 15 \times 0.877^{15+1}]}{(1 - 0.877^{15+1})(1 - 0.877)}$$

$L_2 = 4.902$ entities.

Therefore, the theoretical steady-state mean number of entities in the the two-machine system is

$$\mu_x = L_1 + L_2$$

$\mu_x = 10.263$ entities.

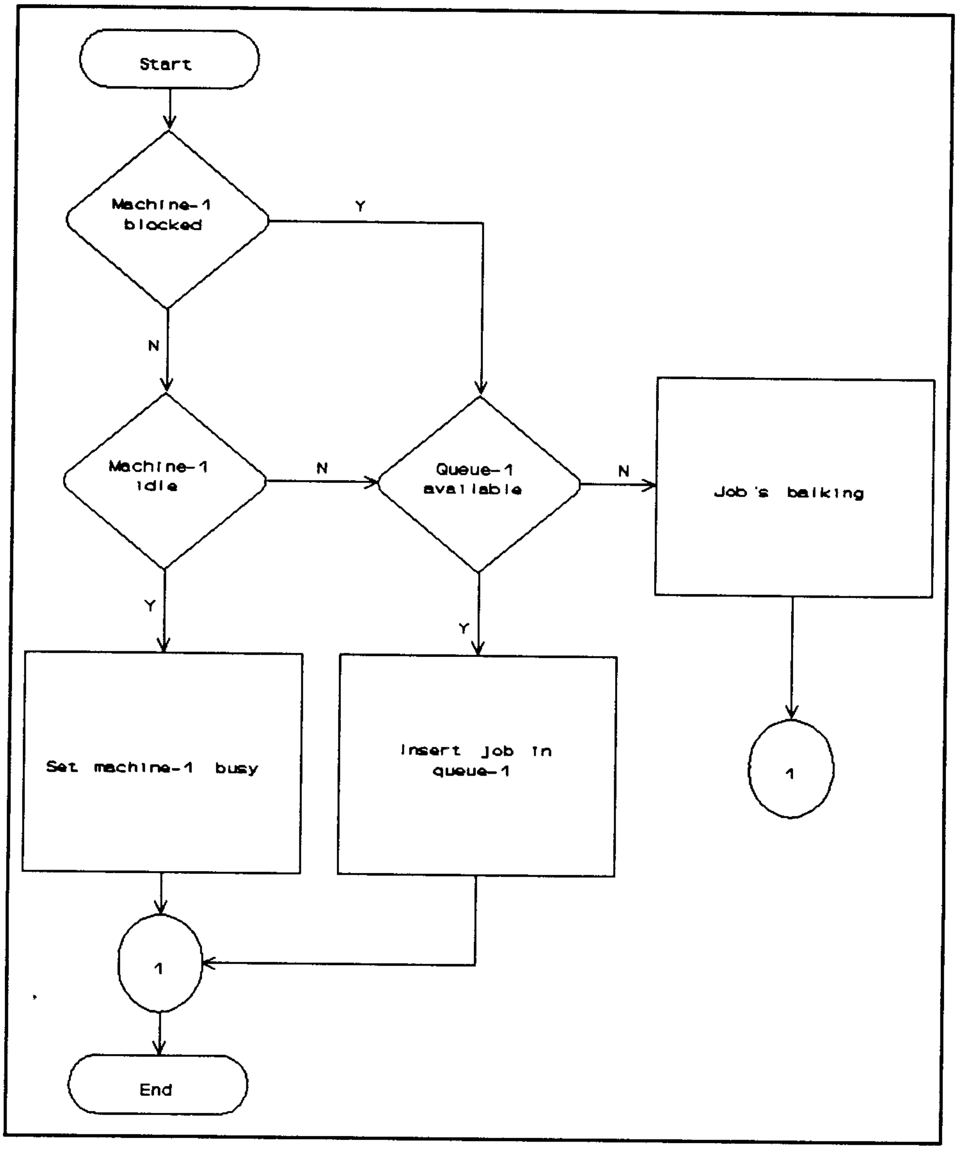
Besides, the average traffic intensity of the system is

$$\overline{\rho} = \frac{\lambda\rho_1 + \overline{\lambda}_2\rho_2}{\lambda + \overline{\lambda}_2}$$

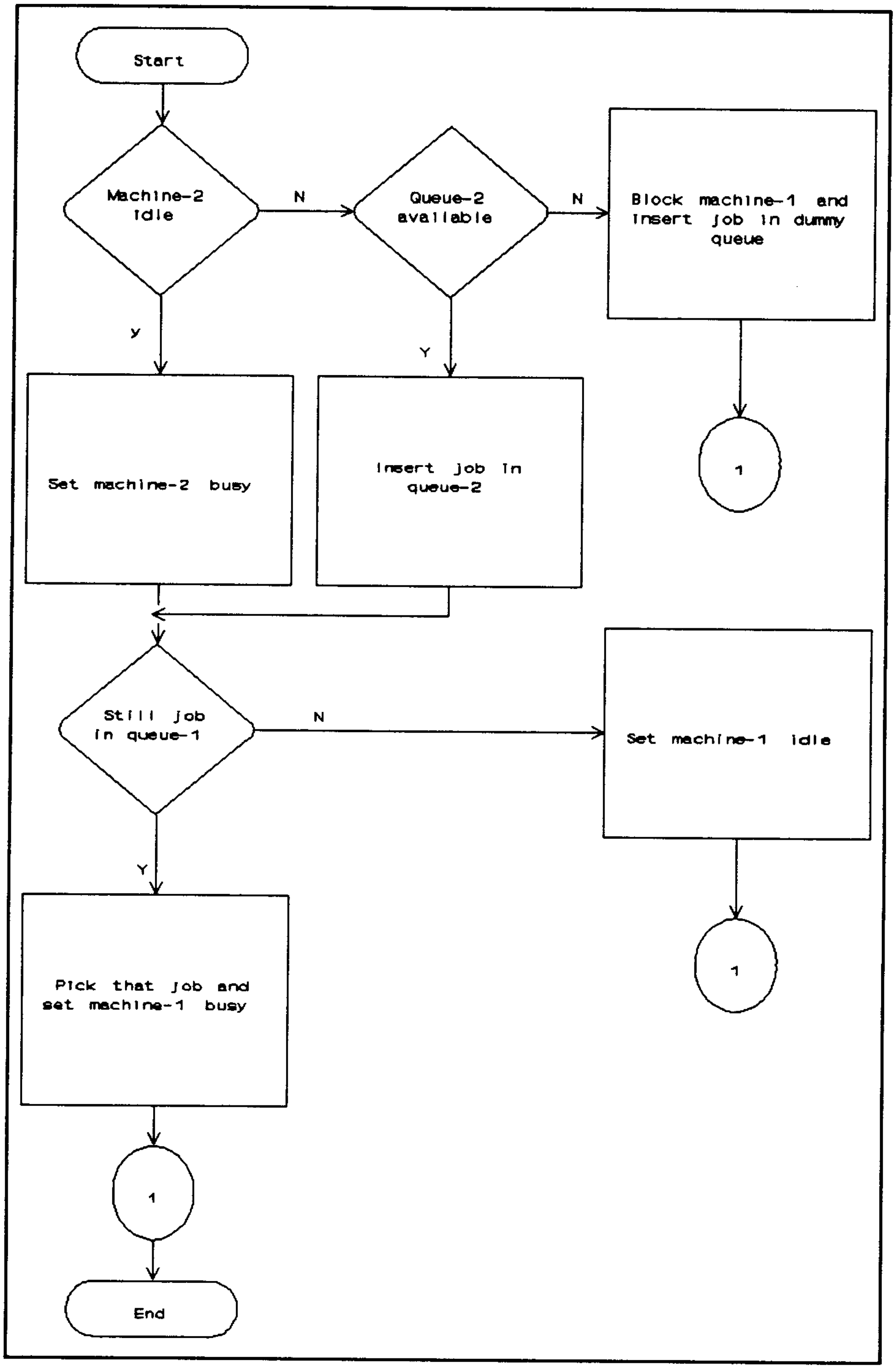
$$\overline{\rho} = \frac{4.5 \times 0.9 + 4.386 \times 0.877}{4.5 + 4.386}$$

$$\overline{\rho} = 0.877.$$

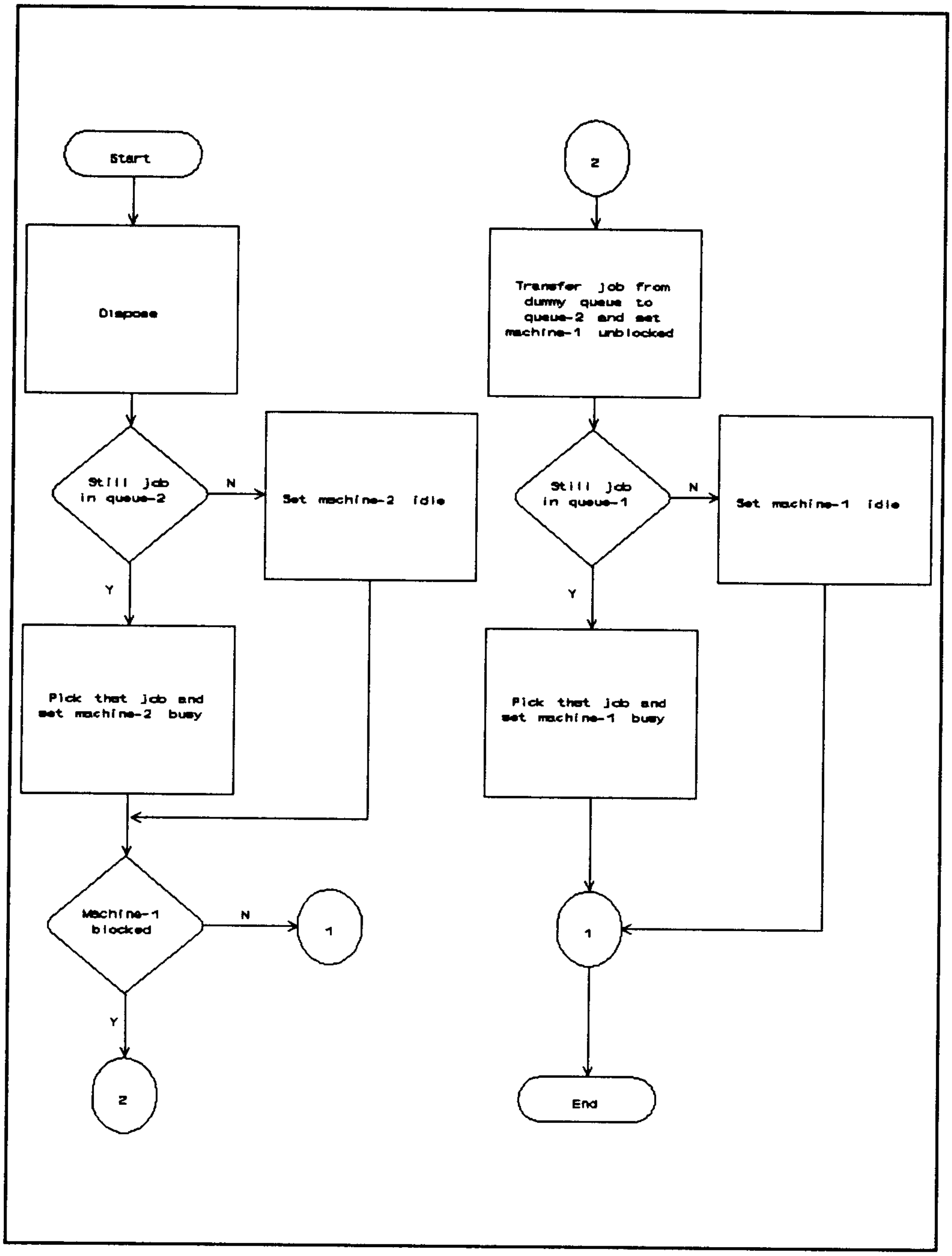
Appendix II-A : Subroutine AR1M1.



Appendix II-B : Subroutine M1TOM2.



Appendix II-C : Subroutine ENDOS.



Appendix III : Theoretical Steady-State Mean Number of Entities in the Three-Machine System.

In order to calculate the theoretical steady-state mean number of entities in the three-machine system (three machines in series or in tandem), the average number of entities calculated separately from each machine must be known first. The average number of entities of the first two machines are already given in Appendix I, that is 5.361 and 4.902 entities.

The arrival rate of entities to the third machine also depends on the effective arrival and service rate as well as on the queue capacity of the second machine. Thus the effective arrival rate of the third machine becomes

$$\bar{\lambda}_3 = \bar{\lambda}_2(1 - P_{(15;2)})$$

where :

$P_{(15;2)}$: probability that an entity will block the second machine if no space is available in the third queue.

$$P_{(15;2)} = \rho_2{}^{c} \; x \; \frac{1 - \rho_2}{1 - \rho_2{}^{c+1}}$$

$$P_{(15;2)} = (\bar{\lambda}_2/\mu)^{c} \; x \; \frac{1 - (\bar{\lambda}_2/\mu)}{1 - (\bar{\lambda}_2/\mu)^{c+1}}$$

$$P_{(15;2)} = 0.877^{15} \; x \; \frac{1 - 0.877}{1 - 0.877^{15+1}}$$

$$P_{(15;2)} = 0.0196.$$

Thus

$$\bar{\lambda}_3 = 4.386(1 - 0.0196)$$

$\bar{\lambda}_3 = 4.300$ arrivals per time unit.

The traffic intensity of the third machine with a service rate (μ) of five entities per time unit is

$$\rho_3 = \bar{\lambda}_3/\mu = 0.860.$$

and the average number of entities in the third machine is

$$L_3 = \frac{0.860[1 - (15+1)0.860^{15} + 15 \ x \ 0.860^{15+1}]}{(1 - 0.860^{15+1})\ (1 - 0.860)}$$

$L_3 = 4.570$ entities.

Therefore, the theoretical steady-state mean number of entities in the the two-machine system is

$$\mu_x = L_1 + L_2 + L_3$$

$\mu_x = 14.833$ entities.

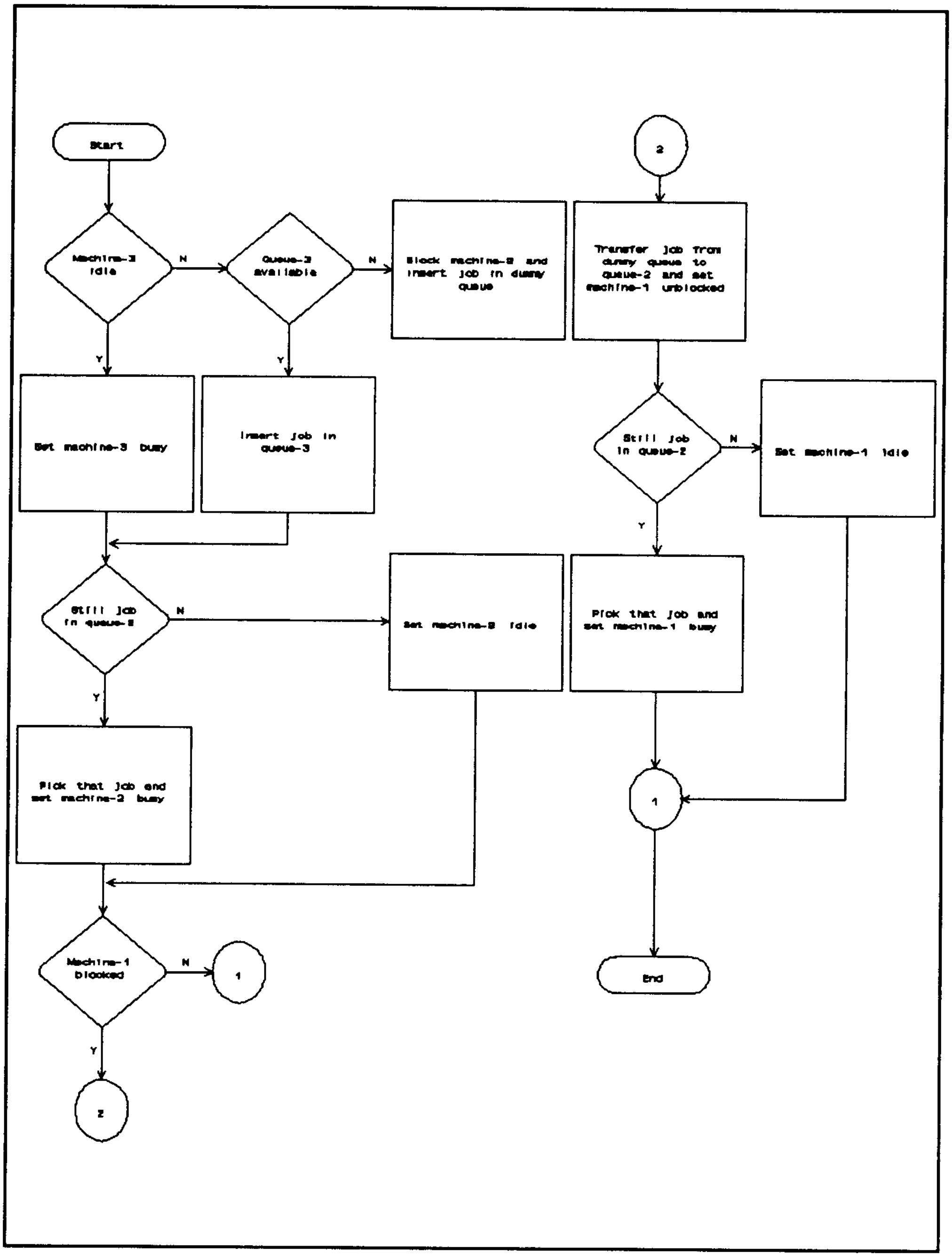
Besides, the average traffic intensity of the system is

$$\bar{\rho} = \frac{\lambda\rho_1 + \bar{\lambda}_2\rho_2 + \bar{\lambda}_3\rho_3}{\lambda + \bar{\lambda}_2 + \bar{\lambda}_3}$$

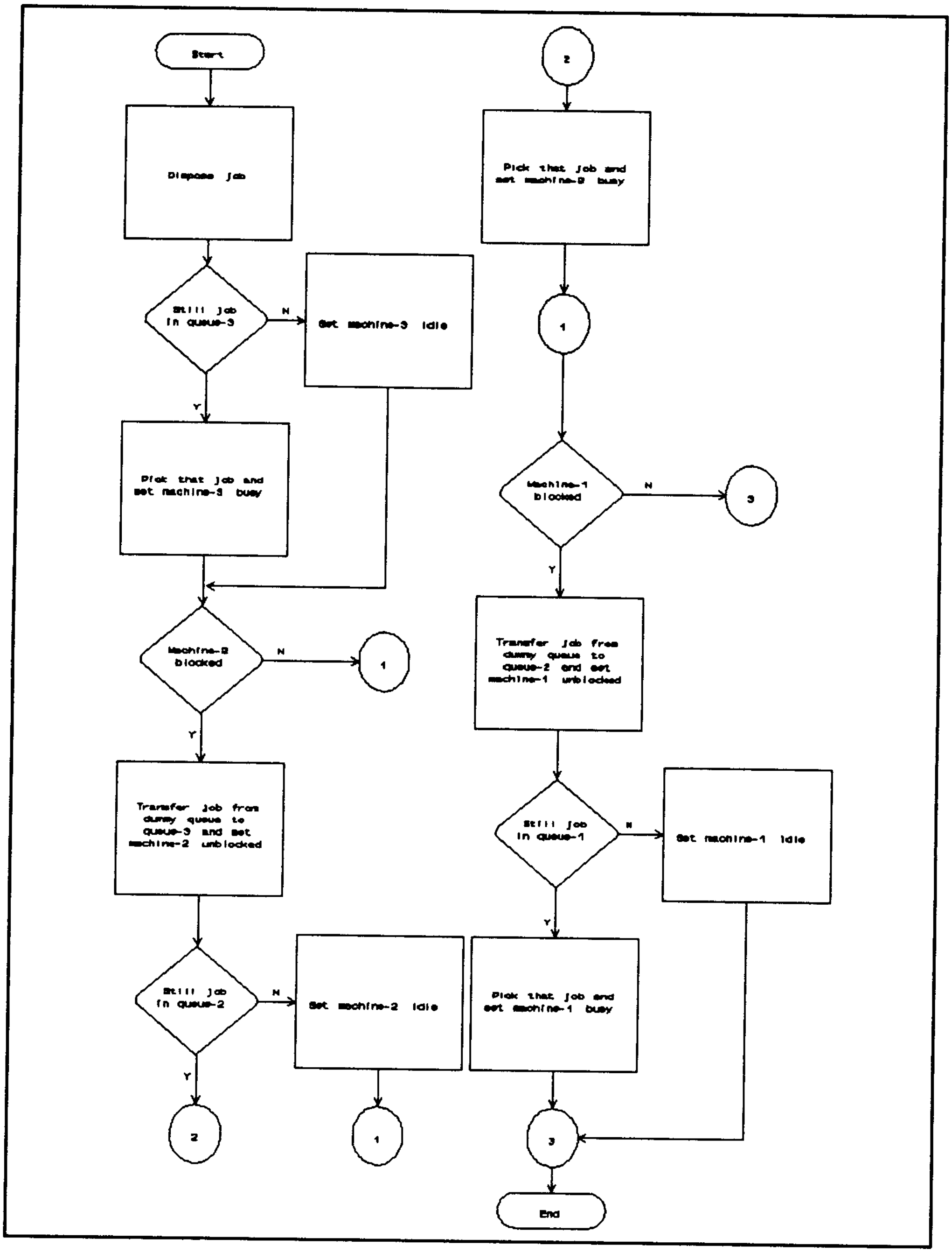
$$\bar{\rho} = \frac{4.5 \ x \ 0.9 + 4.386 \ x \ 0.877 + 4.3 \ x \ 0.86}{4.5 + 4.386 + 4.3}$$

$\bar{\rho} = 0.879.$

Appendix IV-A : Subroutine M2TOM3.



Appendix IV-B : Subroutine ENDOS.



Appendix V-A : Program for the two-machine system.

```
C*****************************************************************C
C     M1M2.FOR : STEADY STATE STUDY WITH 2 MACHINES IN SERIES     C
C                                                                 C
C     Type of Events:                                             C
C       1. Start processing on Machine 1                : ARIM1   C
C       2. Finish on Mach-1, start processing on Mach-2 : M1TOM2  C
C       3. Complete all processes                       : ENDOS   C
C                                                                 C
C     List of Variables :                                         C
C       X(1) : status of Mach-1 -- 0 : idle                       C
C                                  1 : busy                       C
C       X(2) : status of Mach-2 -- 0 : idle                       C
C                                  1 : busy                       C
C       X(3) : blocking status of Mach-1 -- 0 : unblocked         C
C                                           -- 1 : blocked        C
C*****************************************************************C
      SUBROUTINE EVENT(JOB,N)
      GO TO (1,2,3,4), N
1     CALL ARIM1(JOB)
      RETURN
2     CALL M1TOM2(JOB)
      RETURN
3     CALL ENDOS(JOB)
      RETURN
4     CALL INCTIM(JOB)
      RETURN
      END
C
C
      SUBROUTINE PRIME
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,IBLOK,IBALK,CAPQ2,
     2RSUMVAR,RVAR,BIASQR,RMSE
      CALL CREATE(JOB)
      CALL SCHED(JOB,1,EX(1,5))
      CALL CREATE(JOB)
      CALL SCHED(JOB,4,0.0)
C*********************INITIALIZATION*******************************C
      I=0
      X(1)=0.0
      X(2)=0.0
      X(3)=0.0
      NUMSYS=0
      SCSUM=0.0
      RSUMVAR=0.0
      ZEROBS=0.0
      IBALK=0
      IBLOCK=0
C***  Batch size,i.e. number of observations per batch
      IBATCH=CO(3)
C***  Batch count,i.e. number of batch means to be compared
      KBATCH=CO(4)
      K=0
C***  Number of replications required in simulation
      KREP=CO(5)
C***  Number of observations per replications
      NOBS=CO(7)
C***  Queue capacity for Machine 2
```

```
      CAPQ2=CO(9)
C
      DO 160 J=1,KBATCH
      SUMS(J)=0.0
  160 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE ARIM1(NEWJOB)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,IBLOK,IBALK,CAPQ2,
     2RSUMVAR,RVAR,BIASQR,RMSE
C***  For every arrival, increase number in the system by 1
      NUMSYS=NUMSYS+1
C***  Schedule the next arrival in the system
      CALL SCHED(NEWJOB,1,EX(1,5))
C***  Process the current job
      CALL CREATE(JOB)
C
C***  If Machine 1 blocked, check Queue status of Machine 1.
C
      IF (X(3).EQ.1) THEN
        GO TO 170
C
C***  Else, Machine 1 unblocked & idle, set Machine 1 busy.
C     Schedule the completion time.
C
      ELSE
        IF (X(1).EQ.0) THEN
          X(1)=1.0
          CALL SCHED(JOB,2,EX(2,9))
C
C***  Else, Machine 1 unblocked & busy, check Queue status of
C     Machine 1.
C
        ELSE
          GO TO 170
        ENDIF
      ENDIF
      RETURN
C
C***  If Queue of Machine 1 is available, put the job in Queue.
C
  170 IF (NQ(1).LT.15) THEN
        CALL INSERT(JOB,1)
C
C***  Else, no available queue of Machine 1, job is balking.
C
      ELSE
        IBALK=IBALK+1
        CALL DISPOS(JOB)
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE M1TOM2(JOB)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,IBLOK,IBALK,CAPQ2,
```

```
     2RSUMVAR,RVAR,BIASQR,RMSE
C
C***  If Machine 2 idle, process the job on Machine 2.
C     Set Machine 2 busy & schedule the completion time.
C     Check Queue status of Machine 1.
C
      IF (X(2).EQ.0) THEN
        X(2)=1.0
        CALL SCHED(JOB,3,EX(8,3))
        GO TO 180
C
C***  Else, Machine 2 busy & Queue of Machine 2 is available,
C     put the job in Queue for Machine 2.
C     Check Queue status of Machine 1.
C
      ELSE
        IF (NQ(2).LT.CAPQ2) THEN
          CALL INSERT(JOB,2)
          GO TO 180
C
C***  Else, Machine 2 busy & no available Queue of Machine 2,
C     put the blocking job in Q-dummy.
C     Set Machine 1 idle and blocked.
C
        ELSE
          CALL INSERT(JOB,3)
          IBLOCK=IBLOCK+1
          X(1)=0.0
          X(3)=1.0
        ENDIF
      ENDIF
      RETURN
C
C***  If still job in Queue for Machine 1, process the job
C     and schedule the completion time.
C
  180 IF (NQ(1).GT.0) THEN
        JOB=LFR(1)
        CALL REMOVE(JOB,1)
        X(1)=1.0
        CALL SCHED(JOB,2,EX(2,2))
C
C***  Else, no job in Queue for Machine 1, set Machine 1 idle.
C
      ELSE
        X(1)=0.0
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE ENDOS(JOB)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,IBLOK,IBALK,CAPQ2,
     2RSUMVAR,RVAR,BIASQR,RMSE
C
C***  For every completed job, decrease number in the system by 1.
C     Dispose the processed job from the system.
C
      NUMSYS=NUMSYS-1
      CALL DISPOS(JOB)
```

```
C
C***  If still job in queue for Machine 2, process the job and
C     schedule the completion time.
C
      IF (NQ(2).GT.0) THEN
        JOB=LFR(2)
        CALL REMOVE(JOB,2)
        X(2)=1.0
        CALL SCHED(JOB,3,EX(8,3))
C
C***  Else, no job in queue for Machine 2, set Machine 2 idle.
C
      ELSE
        X(2)=0.0
      ENDIF
C
C***  If Machine 1 blocked, transfer blocking job from dummy-queue
C     to Queue of Machine 2. Set Machine 1 unblocked.
C
      IF (X(3).EQ.1) THEN
        JOB=LFR(3)
        CALL REMOVE(JOB,3)
        CALL INSERT(JOB,2)
        X(3)=0.0
C
C***  Else, Machine 1 unblocked. No action taken (Machine 1 could
C     be busy or idle)
C
      ELSE
        RETURN
      ENDIF
C
C***  If still job in Queue for Machine 1, set Machine 1 busy.
C     Process the job and schedule the completion time.
C
      IF (NQ(1).GT.0) THEN
        JOB=LFR(1)
        CALL REMOVE(JOB,1)
        X(1)=1.0
        CALL SCHED(JOB,2,EX(2,2))
C
C***  Else, no job in queue for Machine 1, set Machine 1 idle.
C
      ELSE
        X(1)=0.0
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE INCTIM(JOB)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,IBLOK,IBALK,CAPQ2,
     2RSUMVAR,RVAR,BIASQR,RMSE
C***  I as counter for the ith job, maximum observation is NOBS
      I=I+1
C***  Schedule the next observation according to TBO (Time Between
C***  Observation)
      CALL SCHED(JOB,4,CO(6))
C***  Process the current job
      CALL CREATE(JOB)
```

```
C***  Observe number in the system until NOBS observations
      IF (I.LE.NOBS) THEN
        IOBS(I)=NUMSYS
        IF (IOBS(I).EQ.0) THEN
C***  Count the frequency of zero observations
          ZEROBS=ZEROBS+1
        ENDIF
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE WRAPUP
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,IBLOK,IBALK,CAPQ2,
     2RSUMVAR,RVAR,BIASQR,RMSE
C     CALL COUNT(1,1)
      IS=1
C
C***  Check the batch count, if KBATCH = 2 .... pairwise comparisons ***C
C***                            KBATCH > 2      multiple comparisons ***C
      IF (KBATCH.EQ.2) THEN
C***  Perform the pairwise comparisons
      CALL PAIRCOM
C***  Else, Perform the multiple comparisons
      ELSE
        CALL MULTCOM
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE PAIRCOM
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,IBLOK,IBALK,CAPQ2,
     2RSUMVAR,RVAR,BIASQR,RMSE
C***  Perform the pairwise comparisons
      DO 20 J=1,KBATCH
      DO 10 I=IS,J*IBATCH
      SUMS(J)=IOBS(I)+SUMS(J)
   10 CONTINUE
C***  Mean for each batch
      BMEAN(J)=SUMS(J)/REAL(IBATCH)
      IS=IS+IBATCH
   20 CONTINUE
C***  Difference between those batch means
  100 DIF=ABS(BMEAN(1)-BMEAN(2))
C***  Compare the difference and the epsilon value
      IF (DIF.GT. 4.03) THEN
C
C***  The difference is greater than epsilon value, it is still in
C***  warm-up period and continue to make pairwise comparisons
C***

        IF (IS.GE.NOBS) THEN
          CALL TRUNC
        ELSE
C***
          BMEAN(1)=BMEAN(2)
          SUMS(2)=0.0
          DO 30 I=IS,IS+IBATCH-1
```

```
          SUMS(2)=IOBS(I)+SUMS(2)
   30     CONTINUE
          BMEAN(2)=SUMS(2)/REAL(IBATCH)
          IS=IS+IBATCH
          GO TO 100
C***
        ENDIF
C***
      ELSE
C***  The difference is less than epsilon value, it is already in
C***  steady state period
        CALL TRUNC
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE MULTCOM
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,IBLOK,IBALK,CAPQ2,
     2RSUMVAR,RVAR,BIASQR,RMSE
C***  Perform the multiple comparisons
      DO 50 J=1,KBATCH
      DO 40 I=IS,J*IBATCH
      SUMS(J)=IOBS(I)+SUMS(J)
   40 CONTINUE
C***  Mean for each batch
      BMEAN(J)=SUMS(J)/REAL(IBATCH)
      IS=IS+IBATCH
   50 CONTINUE
C***  Difference between those batch means
  200 DIF1=ABS(BMEAN(1)-BMEAN(2))
      DIF2=ABS(BMEAN(1)-BMEAN(3))
      DIF3=ABS(BMEAN(2)-BMEAN(3))
C***  Select the maximum difference
      DIF=AMAX1(DIF1,DIF2,DIF3)
C***  Compare the maximum difference and the epsilon value
      IF (DIF.GT. 4.03) THEN
C***  The difference is greater than epsilon value, it is still in
C***  warm-up period and continue to make multiple comparisons
C***
        IF (IS.GE.NOBS) THEN
          CALL TRUNC
        ELSE
C***
          BMEAN(1)=BMEAN(2)
          BMEAN(2)=BMEAN(3)
          SUMS(3)=0.0
          DO 60 I=IS,IS+IBATCH-1
          SUMS(3)=IOBS(I)+SUMS(3)
   60     CONTINUE
          BMEAN(3)=SUMS(3)/REAL(IBATCH)
          IS=IS+IBATCH
          GO TO 200
C***
        ENDIF
C***
      ELSE
C***  The difference is less than epsilon value, it is already in
C***  steady state period
        CALL TRUNC
```

```
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE TRUNC
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,IBLOK,IBALK,CAPQ2,
     2RSUMVAR,RVAR,BIASQR,RMSE
C***  Set-up the truncation point
      ITRUNC=IS-1
C***
      IF (ITRUNC.GE.NOBS) THEN
        IS=NOBS+1-IBATCH
      ENDIF
      DO 70 I=IS,NOBS
      SCSUM=REAL(IOBS(I))+SCSUM
   70 CONTINUE
C***  Calculate the Schriber mean for each replication
      SCMEAN=SCSUM/REAL(NOBS-IS+1)
C
C***  Calculate the bias square for each replication
      BIASQR=(SCMEAN-10.2631)**2
C
      DO 80 I=IS,NOBS
      RSUMVAR=RSUMVAR+(REAL(IOBS(I))-SCMEAN)**2
   80 CONTINUE
C***  Calculate the Schriber variance for each replication
      RVAR=RSUMVAR/REAL(NOBS-IS)
C
C***  Calculate MSE for each replication
      RMSE=RVAR+BIASQR
C
C***  Set K to the number of simulation run
      K=NRUN
      IF (K.EQ.1) THEN
C***  Create and open file RESULT.BBB for the first run
        OPEN(UNIT=10,FILE='RESULT.BBB',STATUS='NEW',ACCESS='SEQUENTIAL',
     1  FORM='FORMATTED')
      ENDIF
C***   If # simulation runs is less than or equal the number of
replications
      IF (K.LE.KREP) THEN
C***  Write the desire variables into the file
        WRITE(10,'(F10.4,5X,F10.4,5X,F10.4,5X,F10.4,5X,F10.4,
     1  5X,I5,5X,I5,5X,I5)')SCMEAN,BIASQR,RVAR,RMSE,ZEROBS,
     2  ITRUNC,IBLOCK,IBALK
      ENDIF
C***  If # simulation is equal to the number of replications
      IF (K.EQ.KREP) THEN
C***  Close file RESULT.BBB
        ENDFILE(UNIT=10)
        CLOSE(UNIT=10,STATUS='KEEP')
C***  Call SCSTAT to perform Statistics Calculation
        CALL SCSTAT
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE SCSTAT
```

```
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,IBLOK,IBALK,CAPQ2,
     2RSUMVAR,RVAR,BIASQR,RMSE
      GRSUM=0.0
      SUMSQR=0.0
C***  Open the file and read its value
      OPEN(UNIT=10,FILE='RESULT.BBB',STATUS='OLD',ACCESS='SEQUENTIAL',
     1FORM='FORMATTED')
      DO 90 I=1,KREP
      READ(10,150,END=110)SCMEAN,BIASQR,RVAR,RMSE,ZEROBS,
     1ITRUNC,IBLOCK,IBALK
  150 FORMAT(F10.4,5X,F10.4,5X,F10.4,5X,F10.4,5X,F10.4,5X,I5,5X,I5,
     15X,I5)
      GRSUM=GRSUM+SCMEAN
      SUMSQR=SUMSQR+(SCMEAN**2)
   90 CONTINUE
  110 CLOSE(UNIT=10,STATUS='KEEP')
C***  Calculate the Schriber-mean, bias, and variance for the number
C***  in the system, already in the steady-state condition
      SCAVG=GRSUM/REAL(KREP)
      SCBIAS=SCAVG-10.2631
      VAR=(SUMSQR-(REAL(KREP)*(SCAVG**2)))/REAL(KREP-1)
C***  Calculate the expected MSE
      SMSE=VAR+(SCBIAS**2)
C***  Calculate the Schriber standard error (SE)
      SE=SQRT(VAR/REAL(KREP))
C***  Calculate the Half Length (HL) of the Schriber mean, using 10%
C***  level of confidence (Z = 1.645)
      HL=SE*1.645
      WRITE(*,'(8X,A,I5)') 'Number of batch size :',IBATCH
      WRITE(*,'(8X,A,I5)') 'Number of batch count :',KBATCH
      WRITE(*,'(8X,A,I5)') 'Number of replications :',KREP
      WRITE(*,'(8X,A,F10.4)') 'Time between observations :',CO(6)
      WRITE(*,'(8X,A,F10.4)') 'Schriber grand mean :',SCAVG
      WRITE(*,'(8X,A,F10.4)') 'Schriber bias :',SCBIAS
      WRITE(*,'(8X,A,F10.4)') 'Schriber variance :',VAR
      WRITE(*,'(8X,A,F10.4)') 'Expected MSE :',SMSE
      WRITE(*,'(8X,A,F10.4)') 'Standard Error (se) :',SE
      WRITE(*,'(8X,A,F10.4)') 'Half Length :',HL
      RETURN
      END
```

Appendix V-B : Experimental Frame for two-machine system's program.

```
BEGIN;
PROJECT,TRY,BUDIMAN;
DISCRETE,400,,3;
;Parameters : 1 -- Arrival Rate                  : 0.2222
;             2 -- Service Rate of Machine 1 : 0.2
;             3 -- Batch Size                    : 5 or 10
;             4 -- Batch Count                   : 2
;             5 -- Number of replications        : 1000
;             6 -- Time between observations : 0.25, 0.5,0.75,
;                                                1, 1.25
;             7 -- Number of observations per replication : 50
;             8 -- Service Rate of Machine 2 : 0.2
;             9 -- Capacity of Queue-2           : 15
PARAMETERS:1,0.2222:2,0.2:3,5:4,2:5,1000:6,0.75:7,50:8,0.2:9,15;
REPLICATE,1000,0.0,37.5;
END;
```

Appendix VI-A : Program for the three-machine system.

```
C**************************************************************C
C     M123.FOR : STEADY STATE STUDY WITH 3 MACHINES IN SERIES    C
C                                                                C
C     Type of Events:                                            C
C       1. Start processing on Machine 1                : ARIM1  C
C       2. Finish on Mach-1, start processing on Mach-2 : M1TOM2 C
C       3. Finish on Mach-2, start processing on Mach-3 : M2TOM3 C
C       4. Complete all processes                       : ENDOS  C
C                                                                C
C     List of Variables :                                        C
C       X(1) : status of Mach-1 -- 0 : idle                      C
C                                  1 : busy                      C
C       X(2) : status of Mach-2 -- 0 : idle                      C
C                                  1 : busy                      C
C       X(3) : blocking status of Mach-1 -- 0 : unblocked        C
C                                        -- 1 : blocked          C
C       X(4) : status of Mach-3 -- 0 : idle                      C
C                               -- 1 : busy                      C
C       X(5) : blocking status of Mach-2 -- 0 : unblocked        C
C                                        -- 1 : blocked          C
C**************************************************************C
      SUBROUTINE EVENT(JOB,N)
      GO TO (1,2,3,4,5), N
1     CALL ARIM1(JOB)
      RETURN
2     CALL M1TOM2(JOB)
      RETURN
3     CALL M2TOM3(JOB)
      RETURN
4     CALL ENDOS(JOB)
      RETURN
5     CALL INCTIM(JOB)
      RETURN
      END
C
C
      SUBROUTINE PRIME
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,CAPQ2,CAPQ3,
     2RSUMVAR,RVAR,BIASQR,RMSE
      CALL CREATE(JOB)
      CALL SCHED(JOB,1,EX(1,5))
      CALL CREATE(JOB)
      CALL SCHED(JOB,5,0.0)
C*********************INITIALIZATION****************************C
      I=0
      DO 210 L=1,5
      X(L)=0.0
  210 CONTINUE
      NUMSYS=0
      SCSUM=0.0
      RSUMVAR=0.0
      ZEROBS=0.0
C***  Batch size,i.e. number of observations per batch
      IBATCH=CO(3)
C***  Batch count,i.e. number of batch means to be compared
      KBATCH=CO(4)
      K=0
C***  Number of replications required in simulation
```

```
      KREP=CO(5)
C***  Number of observations per replications
      NOBS=CO(7)
C***  Queue capacity for Machine 2
      CAPQ2=CO(9)
C***  Queue capacity for Machine 3
      CAPQ3=CO(11)
C
      DO 160 J=1,KBATCH
      SUMS(J)=0.0
  160 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE ARIM1(NEWJOB)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,CAPQ2,CAPQ3,
     2RSUMVAR,RVAR,BIASQR,RMSE
C***  For every arrival, increase number in the system by 1
      NUMSYS=NUMSYS+1
C***  Schedule the next arrival in the system
      CALL SCHED(NEWJOB,1,EX(1,5))
C***  Process the current job
      CALL CREATE(JOB)
C
C***  If Machine 1 blocked, check Queue status of Machine 1.
C
      IF (X(3).EQ.1) THEN
        GO TO 170
C
C***  Else, Machine 1 unblocked & idle, set Machine 1 busy.
C     Schedule the completion time.
C
      ELSE
        IF (X(1).EQ.0) THEN
          X(1)=1.0
          CALL SCHED(JOB,2,EX(2,2))
C
C***  Else, Machine 1 unblocked & busy, check Queue status of
C     Machine 1.
C
        ELSE
          GO TO 170
        ENDIF
      ENDIF
      RETURN
C
C***  If Queue of Machine 1 is available, put the job in Queue.
C
  170 IF (NQ(1).LT.15) THEN
        CALL INSERT(JOB,1)
C
C***  Else, no available queue of Machine 1, job is balking.
C
      ELSE
        CALL DISPOS(JOB)
      ENDIF
      RETURN
      END
C
```

```
C
      SUBROUTINE M1TOM2(JOB)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,CAPQ2,CAPQ3,
     2RSUMVAR,RVAR,BIASQR,RMSE
C
C***  If Machine 2 idle, process the job on Machine 2.
C     Set Machine 2 busy & schedule the completion time.
C     Check Queue status of Machine 1.
C
      IF (X(2).EQ.0) THEN
        X(2)=1.0
        CALL SCHED(JOB,3,EX(8,3))
        GO TO 180
C
C***  Else, Machine 2 busy & Queue of Machine 2 is available,
C     put the job in Queue for Machine 2.
C     Check Queue status of Machine 1.
C
      ELSE
        IF (NQ(2).LT.CAPQ2) THEN
          CALL INSERT(JOB,2)
          GO TO 180
C
C***  Else, Machine 2 busy & no available Queue of Machine 2,
C     put the blocking job in Q-dummy.
C     Set Machine 1 idle and blocked.
C
        ELSE
          CALL INSERT(JOB,3)
          X(1)=0.0
          X(3)=1.0
        ENDIF
      ENDIF
      RETURN
C
C***  If still job in Queue for Machine 1, process the job
C     and schedule the completion time.
C
  180 IF (NQ(1).GT.0) THEN
        JOB=LFR(1)
        CALL REMOVE(JOB,1)
        X(1)=1.0
        CALL SCHED(JOB,2,EX(2,2))
C
C***  Else, no job in Queue for Machine 1, set Machine 1 idle.
C
      ELSE
        X(1)=0.0
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE M2TOM3(JOB)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,CAPQ2,CAPQ3,
     2RSUMVAR,RVAR,BIASQR,RMSE
C
C***  If Machine 3 idle, process the job on Machine 3.
```

```
C     Set Machine 3 busy & schedule the completion time.
C
      IF (X(4).EQ.0) THEN
        X(4)=1.0
        CALL SCHED(JOB,4,EX(10,5))
C
C***  Else, Machine 3 busy & Queue of Machine 3 is available,
C     put the job in Queue for Machine 3.
C
      ELSE
        IF (NQ(4).LT.CAPQ3) THEN
          CALL INSERT(JOB,4)
C
C***  Else, Machine 3 busy & no available Queue of Machine 3,
C     put the blocking job in Q-dummy.
C     Set Machine 2 idle and blocked.
C
        ELSE
          CALL INSERT(JOB,5)
          X(2)=0.0
          X(5)=1.0
          RETURN
        ENDIF
      ENDIF
C
C***  If still job in queue for Machine 2, process the job and
C     schedule the completion time.
C
      IF (NQ(2).GT.0) THEN
        JOB=LFR(2)
        CALL REMOVE(JOB,2)
        X(2)=1.0
        CALL SCHED(JOB,3,EX(8,3))
C
C***  Else, no job in queue for Machine 2, set Machine 2 idle.
C
      ELSE
        X(2)=0.0
      ENDIF
C
C***  If Machine 1 blocked, transfer blocking job from dummy-queue
C     to Queue of Machine 2. Set Machine 1 unblocked.
C
      IF (X(3).EQ.1) THEN
        JOB=LFR(3)
        CALL REMOVE(JOB,3)
        CALL INSERT(JOB,2)
        X(3)=0.0
C
C***  Else, Machine 1 unblocked. No action taken (Machine 1 could
C     be busy or idle)
C
      ELSE
        RETURN
      ENDIF
C
C***  If still job in Queue for Machine 1, set Machine 1 busy.
C     Process the job and schedule the completion time.
C
      IF (NQ(1).GT.0) THEN
        JOB=LFR(1)
        CALL REMOVE(JOB,1)
```

```
          X(1)=1.0
          CALL SCHED(JOB,2,EX(2,2))
C
C***   Else, no job in queue for Machine 1, set Machine 1 idle.
C
        ELSE
          X(1)=0.0
        ENDIF
        RETURN
        END
C
C
        SUBROUTINE ENDOS(JOB)
        COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
        COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
       1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,CAPQ2,CAPQ3,
       2RSUMVAR,RVAR,BIASQR,RMSE
C
C***   For every completed job, decrease number in the system by 1.
C      Dispose the processed job from the system.
C
        NUMSYS=NUMSYS-1
        CALL DISPOS(JOB)
C
C***   If still job in queue for Machine 3, process the job and
C      schedule the completion time.
C
        IF (NQ(4).GT.0) THEN
          JOB=LFR(4)
          CALL REMOVE(JOB,4)
          X(4)=1.0
          CALL SCHED(JOB,4,EX(10,5))
C
C***   Else, no job in queue for Machine 3, set Machine 3 idle.
C
        ELSE
          X(4)=0.0
        ENDIF
C
C***   If Machine 2 blocked, transfer blocking job from dummy-queue
C      to Queue of Machine 3. Set Machine 2 unblocked.
C
        IF (X(5).EQ.1) THEN
          JOB=LFR(5)
          CALL REMOVE(JOB,5)
          CALL INSERT(JOB,4)
          X(5)=0.0
C
C***   If still job in queue for Machine 2, process the job and
C      schedule the completion time.
C
          IF (NQ(2).GT.0) THEN
            JOB=LFR(2)
            CALL REMOVE(JOB,2)
            X(2)=1.0
            CALL SCHED(JOB,3,EX(8,3))
C
C***   Else, no job in queue for Machine 2, set Machine 2 idle.
C
          ELSE
            X(2)=0.0
          ENDIF
```

```
C
C***  Else, Machine 2 unblocked. No action taken (Machine 2 could
C     be busy or idle)
C
      ENDIF
C
C***  If Machine 1 blocked, transfer blocking job from dummy-queue
C     to Queue of Machine 2. Set Machine 1 unblocked.
C
      IF (X(3).EQ.1) THEN
        JOB=LFR(3)
        CALL REMOVE(JOB,3)
        CALL INSERT(JOB,2)
        X(3)=0.0
C
C***  If still job in Queue for Machine 1, set Machine 1 busy.
C     Process the job and schedule the completion time.
C
        IF (NQ(1).GT.0) THEN
          JOB=LFR(1)
          CALL REMOVE(JOB,1)
          X(1)=1.0
          CALL SCHED(JOB,2,EX(2,2))
C
C***  Else, no job in queue for Machine 1, set Machine 1 idle.
C
        ELSE
          X(1)=0.0
        ENDIF
C
C***  Else, Machine 1 unblocked. No action taken (Machine 1 could
C     be busy or idle)
C
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE INCTIM(JOB)
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,CAPQ2,CAPQ3,
     2RSUMVAR,RVAR,BIASQR,RMSE
C***  I as counter for the ith job, maximum observation is NOBS
      I=I+1
C***  Schedule the next observation according to TBO (Time Between
C***  Observation)
      CALL SCHED(JOB,5,CO(6))
C***  Process the current job
      CALL CREATE(JOB)
C***  Observe number in the system until NOBS observations
      IF (I.LE.NOBS) THEN
        IOBS(I)=NUMSYS
        IF (IOBS(I).EQ.0) THEN
C***  Count the frequency of zero observations
          ZEROBS=ZEROBS+1
        ENDIF
      ENDIF
      RETURN
      END
C
C
```

```
      SUBROUTINE WRAPUP
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,CAPQ2,CAPQ3,
     2RSUMVAR,RVAR,BIASQR,RMSE
C     CALL COUNT(1,1)
      IS=1
C
C***  Check the batch count, if KBATCH = 2 .... pairwise comparisons ***C
C***                             KBATCH > 2      multiple comparisons ***C
      IF (KBATCH.EQ.2) THEN
C***  Perform the pairwise comparisons
      CALL PAIRCOM
C***  Else, Perform the multiple comparisons
      ELSE
        CALL MULTCOM
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE PAIRCOM
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,CAPQ2,CAPQ3,
     2RSUMVAR,RVAR,BIASQR,RMSE
C***  Perform the pairwise comparisons
      DO 20 J=1,KBATCH
      DO 10 I=IS,J*IBATCH
      SUMS(J)=IOBS(I)+SUMS(J)
   10 CONTINUE
C***  Mean for each batch
      BMEAN(J)=SUMS(J)/REAL(IBATCH)
      IS=IS+IBATCH
   20 CONTINUE
C***  Difference between those batch means
  100 DIF=ABS(BMEAN(1)-BMEAN(2))
C***  Compare the difference and the epsilon value
      IF (DIF.GT. 4.03) THEN
C
C***  The difference is greater than epsilon value, it is still in
C***  warm-up period and continue to make pairwise comparisons
C***
        IF (IS.GE.NOBS) THEN
          CALL TRUNC
        ELSE
C***
          BMEAN(1)=BMEAN(2)
          SUMS(2)=0.0
          DO 30 I=IS,IS+IBATCH-1
          SUMS(2)=IOBS(I)+SUMS(2)
   30     CONTINUE
          BMEAN(2)=SUMS(2)/REAL(IBATCH)
          IS=IS+IBATCH
          GO TO 100
C***
        ENDIF
C***
      ELSE
C***  The difference is less than epsilon value, it is already in
C***  steady state period
        CALL TRUNC
```

```
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE MULTCOM
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,CAPQ2,CAPQ3,
     2RSUMVAR,RVAR,BIASQR,RMSE
C***  Perform the multiple comparisons
      DO 50 J=1,KBATCH
      DO 40 I=IS,J*IBATCH
      SUMS(J)=IOBS(I)+SUMS(J)
   40 CONTINUE
C***  Mean for each batch
      BMEAN(J)=SUMS(J)/REAL(IBATCH)
      IS=IS+IBATCH
   50 CONTINUE
C***  Difference between those batch means
  200 DIF1=ABS(BMEAN(1)-BMEAN(2))
      DIF2=ABS(BMEAN(1)-BMEAN(3))
      DIF3=ABS(BMEAN(2)-BMEAN(3))
C***  Select the maximum difference
      DIF=AMAX1(DIF1,DIF2,DIF3)
C***  Compare the maximum difference and the epsilon value
      IF (DIF.GT. 4.03) THEN
C***  The difference is greater than epsilon value, it is still in
C***  warm-up period and continue to make multiple comparisons
C***
        IF (IS.GE.NOBS) THEN
          CALL TRUNC
        ELSE
C***
          BMEAN(1)=BMEAN(2)
          BMEAN(2)=BMEAN(3)
          SUMS(3)=0.0
          DO 60 I=IS,IS+IBATCH-1
          SUMS(3)=IOBS(I)+SUMS(3)
   60     CONTINUE
          BMEAN(3)=SUMS(3)/REAL(IBATCH)
          IS=IS+IBATCH
          GO TO 200
C***
        ENDIF
C***
      ELSE
C***  The difference is less than epsilon value, it is already in
C***  steady state period
        CALL TRUNC
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE TRUNC
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,CAPQ2,CAPQ3,
     2RSUMVAR,RVAR,BIASQR,RMSE
C***  Set-up the truncation point
      ITRUNC=IS-1
```

```
C***
      IF (ITRUNC.GE.NOBS) THEN
        IS=NOBS+1-IBATCH
      ENDIF
      DO 70 I=IS,NOBS
      SCSUM=REAL(IOBS(I))+SCSUM
   70 CONTINUE
C***  Calculate the Schriber mean for each replication
      SCMEAN=SCSUM/REAL(NOBS-IS+1)
C
C***  Calculate the bias square for each replication
      BIASQR=(SCMEAN-10.2631)**2
C
      DO 80 I=IS,NOBS
      RSUMVAR=RSUMVAR+(REAL(IOBS(I))-SCMEAN)**2
   80 CONTINUE
C***  Calculate the Schriber variance for each replication
      RVAR=RSUMVAR/REAL(NOBS-IS)
C
C***  Calculate MSE for each replication
      RMSE=RVAR+BIASQR
C
C***  Set K to the number of simulation run
      K=NRUN
      IF (K.EQ.1) THEN
C***  Create and open file RESULT.BBB for the first run
        OPEN(UNIT=10,FILE='RESULT.BBB',STATUS='NEW',ACCESS='SEQUENTIAL',
     1  FORM='FORMATTED')
      ENDIF
C***    If # simulation runs is less than or equal the number of
replications
      IF (K.LE.KREP) THEN
C***  Write the desire variables into the file
        WRITE(10,'(F10.4,5X,F10.4,5X,F10.4,5X,F10.4,5X,F10.4,
     1  5X,I5)')SCMEAN,BIASQR,RVAR,RMSE,ZEROBS,ITRUNC
      ENDIF
C***  If # simulation is equal to the number of replications
      IF (K.EQ.KREP) THEN
C***  Close file RESULT.BBB
        ENDFILE(UNIT=10)
        CLOSE(UNIT=10,STATUS='KEEP')
C***  Call SCSTAT to perform Statistics Calculation
        CALL SCSTAT
      ENDIF
      RETURN
      END
C
C
      SUBROUTINE SCSTAT
      COMMON/SIM/D(50),DL(50),S(50),SL(50),X(50),DTNOW,TNOW,TFIN,J,NRUN
      COMMON/USER1/IOBS(300),I,IS,IBATCH,KBATCH,NUMSYS,SUMS(3),BMEAN(3),
     1ITRUNC,SCSUM,ZEROBS,SCMEAN,K,KREP,NOBS,CAPQ2,CAPQ3,
     2RSUMVAR,RVAR,BIASQR,RMSE
      GRSUM=0.0
      SUMSQR=0.0
C***  Open the file and read its value
      OPEN(UNIT=10,FILE='RESULT.BBB',STATUS='OLD',ACCESS='SEQUENTIAL',
     1FORM='FORMATTED')
      DO 90 I=1,KREP
      READ(10,150,END=110)SCMEAN,BIASQR,RVAR,RMSE,ZEROBS,ITRUNC
  150 FORMAT(F10.4,5X,F10.4,5X,F10.4,5X,F10.4,5X,F10.4,5X,I5)
      GRSUM=GRSUM+SCMEAN
```

```
      SUMSQR=SUMSQR+(SCMEAN**2)
   90 CONTINUE
  110 CLOSE(UNIT=10,STATUS='KEEP')
C***  Calculate the Schriber-mean, bias, and variance for the number
C***  in the system, already in the steady-state condition
      SCAVG=GRSUM/REAL(KREP)
      SCBIAS=SCAVG-14.8333
      VAR=(SUMSQR-(REAL(KREP)*(SCAVG**2)))/REAL(KREP-1)
C***  Calculate the expected MSE
      SMSE=VAR+(SCBIAS**2)
C***  Calculate the Schriber standard error (SE)
      SE=SQRT(VAR/REAL(KREP))
C***  Calculate the Half Length (HL) of the Schriber mean, using 10%
C***  level of confidence (Z = 1.645)
      HL=SE*1.645
      WRITE(*,'(8X,A,I5)') 'Number of batch size :',IBATCH
      WRITE(*,'(8X,A,I5)') 'Number of batch count :',KBATCH
      WRITE(*,'(8X,A,I5)') 'Number of replications :',KREP
      WRITE(*,'(8X,A,F10.4)') 'Time between observations :',CO(6)
      WRITE(*,'(8X,A,F10.4)') 'Schriber grand mean :',SCAVG
      WRITE(*,'(8X,A,F10.4)') 'Schriber bias :',SCBIAS
      WRITE(*,'(8X,A,F10.4)') 'Schriber variance :',VAR
      WRITE(*,'(8X,A,F10.4)') 'Expected MSE :',SMSE
      WRITE(*,'(8X,A,F10.4)') 'Standard Error (se) :',SE
      WRITE(*,'(8X,A,F10.4)') 'Half Length :',HL
      RETURN
      END
```

Appendix VI-B : Experimental Frame for the three-machine system's program.

```
BEGIN;
PROJECT,M123,BUDIMAN;
DISCRETE,450,,5;
;Parameters : 1 -- Arrival Rate                : 0.2222
;             2 -- Service Rate of Machine 1   : 0.2
;             3 -- Batch Size                  : 5 or 10
;             4 -- Batch Count                 : 2
;             5 -- Number of replications      : 1000
;             6 -- Time between observations   : 0.375,0.5,0.625
;             7 -- Number of observations per replication : 50
;             8 -- Service Rate of Machine 2   : 0.2
;             9 -- Capacity of Queue-2         : 15
;            10 -- Service Rate of Machine 3   : 0.2
;            11 -- Capacity of Queue-3         : 15
PARAMETERS:1,0.2222:2,0.2:3,5:4,2:5,3:6,0.625:7,20:8,0.2:9,15:
           10,0.2:11,15;
REPLICATE,3,0.0,12.5;
END;
```