THE BEHAVIOR OF TRANSIENT PERIOD OF NONTERMINATING SIMULATIONS: AN EXPERIMENTAL ANALYSIS

A THESIS

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BİLKENT UNIVESITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Burhaneddin Sandıkçı

August, 2001

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis of the degree of Master of Science

Assoc. Prof. İhsan Sabuncuoğlu (Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis of the degree of Master of Science

Asst. Prof. Oya Ekin Karaşan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis of the degree of Master of Science

Asst. Prof. Yavuz Günalay

Approved for the Institute of Engineering and Sciences

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

The design and control of many industrial and service systems require the analysts to account for uncertainty. Computer simulation is a frequently used technique for analyzing uncertain (or stochastic) systems. One disadvantage of simulation modeling is that simulation results are only estimates of model performance measures. Therefore, to obtain better estimates, the outputs of a simulation run should undergo a careful statistical analysis. Simulation studies can be classified as terminating and nonterminating according to the output analysis techniques used. One of the major problems in the output analysis of nonterminating simulations is the problem of initial transient. This problem arises due to initializing simulation runs in an unrepresentative state of the steady-state conditions.

Many techniques have been proposed in the literature to deal with the problem of initial transient. However, existing studies try to improve the efficiency and effectiveness of currently proposed techniques. No research has been encountered that analyzes the behavior of the transient period. In this thesis, we investigate the factors affecting the length of the transient period for nonterminating manufacturing simulations, particularly for serial production lines and job-shop production systems. Factors such as variability of processing times, system size, existence of bottleneck, reliability of system, system load level, and buffer capacity are investigated.

Keywords: Nonterminating simulations, behavior of transient period, serial production lines, job-shop systems, MSER heuristics.

ÖZET

Birçok endüstriyel ve servis sisteminin tasarım ve kontrolü için analizi yapan kimselerin berlirsizliği hesaba katlamaları gerekir. Bilgisayar benzetim tekniği belirsiz (veya rassal) sistemlerin analizinde sıkça kullanılan bir yöntemdir. Benzetim modellerinin önemli bir eksiği performans değerleri için sadece tahminler üretmesidir. Bu nedenle, daha doğru sonuçlar elde edebilmek için benzetim çıktıları dikkatli istatistiki analize tabi tutulmalıdır. Benzetim çalışmaları, kullanılan çıktı analizi tekniklerine göre sonu belirli ve sonu belirsiz olarak sınıflandırılabilirler. Sonu belirsiz benzetimlerin çıktı analizinde karşılaşılan en önemli problemlerden biri geçiş dönemi problemidir. Bu problem benzetim modelini uzun vadedeki durumundan uzak bir konumda başlatmaktan ötürü ortaya çıkar.

Başlangıçtaki geçiş dönemi probleminin çözümüne dair önerilmiş pekçok teknik literatürde mevcuttur. Fakat mevcut çalışmalar daha çok önerilen tekniklerin etkinlik ve yeterliliğini geliştirmeye çalışmaktadır. Geçiş döneminin davranışını inceleyen bir çalışma ile karşılaşılmamıştır. Biz bu tezde, sonu belirsiz benzetimlerle analiz edilen imalat sistemlerinin geçiş dönemini etkileyen faktörleri inceliyoruz. Özellikle seri üretim hatları ve atölye sistemleri üzerinde duruyoruz. İşlem zamanının değişkenliği, sistemin büyüklüğü, darboğazın mevcudiyeti, sistemin güvenilirliği, sistemin yük seviyesi ve tamponların kapasitesi incelenen faktörler arasındadır.

Anahtar Sözcükler: Sonu belirsiz benzetim, geçiş dönemi davranışı, seri üretim hatları, atölyeler, MSER bulgusalları.

to my family

ACKNOWLEDGEMENTS

When I approached the near end, I recognized that it was not a so long journey. However, while I was in the middle of this suffering process, it seemed to be never ending. Thanks God for the knowledge, patience, and power He gave to me to finish this and all other tasks.

I would like to express my sincere gratitude to Assoc. Prof. İhsan Sabuncuoğlu for his supervision and encouragement during my graduate study. His trust and understanding motivated me and let this thesis come to an end.

I am indebted to Asst. Prof. Oya Ekin Karaşan for her invaluable support during my first year at Bilkent University. I owe so much to her for her interest and encouragement. I also would like to thank her for reading and reviewing this thesis.

I am greateful to Asst. Prof. Yavuz Günalay for accepting to read and review this thesis and for his suggestions.

I am so much indebted to Prof. Kemal Sandıkçı for all his invaluable teachings and for introducing me to the One upon whom the universe is created for. I also would like to thank to Prof. Erkan Türe for his never ending morale support during all my studies. Special thanks goes to Prof. K. Preston White for his technical support about MSER heuristics.

I have so much enjoyed meeting Abdullah Karaman and he deserves very special thanks for being a so good friend. Thanks to Mustafa Akbas for his technical help and patience. I also would like thank Çağrı Gürbüz, Onur Boyabatlı, Savaş Çevik, Rabia Köylü, Ayten Türkcan, Banu Yüksel, Erkan Köse, and Hüseyin Karakuş for their friendship.

Ali Kasap, Erkan Sarıkaya, Murat Ay, Ersin Yurdakul, Berkant Çamdereli, and Almula Şahin all shared my troubles and helped me in those days. I would not be in my current situation without their guidance.

Finally, I would like to express my deepest gratitude to Zahide Yavuz for her patience, understanding, kindness, and the happiness she brought to my life.

Contents

List of Figures

List of Tables

1 INTRODUCTION

The need for modeling

The idea of conceptualization and modeling plays a crucial role in our understanding of our environment. The modeling concept is often used to represent and express ideas and objects.

A *model* is defined as *a* representation of a *system* for the purpose of studying the system. In other words, models are used because there is a need for learning something about a system, such as relationships among various components, or performance under some specified conditions, which cannot be observed or experimented directly either due to non-existence of the system or difficulty in manipulation of the system. A carefully built model can throw away the complexity and leave only the necessary parts that an analyst is looking for.

The word "system" comprises the vital part of the definition of a model. A *system* is defined as a group of objects that are joined together in some regular interaction or interdependence toward the accomplishment of some purpose (Banks et al., 1996). The relationships among these objects and the manner in which they interact determine how the system behaves and how well it fulfills its overall purpose. If it is possible and cost-effective to alter the system physically, then it is probably desirable to do so rather than working with a model. However, it is rarely feasible to do this, because such experimentation would often be too costly or too disruptive to the system. Moreover, the system might not even exist, but it can be of interest to study it in its various proposed alternative configurations to see how it should be built in the first place. For these reasons, it is usually necessary to build a model as a representation of the system and study it as a surrogate for the actual system.

Simulation modeling

Various kinds of models can be built to study systems, where our focus is only on mathematical models. Mathematical models represent a system in terms of logical and quantitative relationships. Mathematical models can be grouped into two categories, namely, analytical models and simulation models. *Analytical models* make use of mathematical methods, such as algebra, calculus, or probability theory, which obtain exact information on the questions of interest. On the other hand, *simulation models* evaluate a system numerically and produce only *estimates* of the true characteristics of the system.

A formal definition of simulation by Fishman (1978) is given as follows: *"Simulation is the creation of a model that imitates the behavior of a system, running the model to generate observations of this behavior, and analyzing the observations to understand and summarize this behavior."*

If the model, hence the system is simple enough, then analytical solutions may be possible. If an analytical solution to a model is possible and computationally efficient, it is usually desirable to study the model in this way rather than via a simulation. However, most real-world systems are too complex to allow realistic models to be evaluated analytically, and simulation takes the first place.

Simulation models can be static or dynamic according to the involvement of the passage of time. Another classification of simulation models is on the basis of the characteristics of the input components. If all the inputs are constants than it is called deterministic simulation, whereas a model that contains at least one random input is called stochastic simulation. Stochastic simulation models produce output that is itself random, and therefore must be treated only as an estimate of the true characteristic of the model. A further classification can be made with regard to the occurrence of events. A discrete-event simulation concerns the modeling of a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time. A continuous simulation concerns the modeling of a system over time by a representation in which the state variables change continuously with respect to time (Law and Kelton, 2000). In this study, we build discrete, dynamic, and stochastic models. The steps needed for successful application of a simulation study, and advantages/disadvantages of simulation are discussed well in Banks et al. (1996) and Law and Kelton (2000).

There are many application areas of simulation where it is used extensively in the design and analysis of manufacturing systems due to the high degree of complexity. In this thesis, we will focus only on manufacturing system simulations, particularly, serial production lines and job-shop production systems. Other application areas are well listed in Banks et al. (1996) and Winter Simulation Conference (WSC) proceedings are good sources to find interesting applications.

The need for analysis of simulation outputs

From the definition of simulation given above, it can be seen that simulation is just a computerized experiment of a model. Since random samples from probability distributions are typically used as inputs to simulation experiments, the outputs of the experiments will clearly be random variables, too. And the outputs (*or* estimates) are just particular realizations of random variables that may have large variances. As a result, these estimates could, in a particular run, differ greatly from the corresponding true characteristics of the model. Therefore, there may be a significant probability of making faulty inferences about the system under study. In order to correctly interpret the results of such an experiment, it is necessary to use appropriate statistical analysis tools.

Two crucial problems with an output sequence obtained from a single simulation run are the nonstationarity and the autocorrelation. *Nonstationarity* means that the distributions of the successive observations change over time, whereas *autocorrelation* means the observations in the sequence are correlated with each other. Unless carefully analyzed, these two problems may lead the analyst to wrong conclusions.

The problem of initial transient in simulation outputs

Simulation experiments can be classified as either terminating or nonterminating when the output analysis methods are concerned (Law and Kelton, 2000; Fishman, 1978). A *terminating simulation* is the one for which the starting and stopping conditions are determined *a priori* by reasoning from the underlying system. Although the starting and stopping conditions are determined by the analyst, which in fact is decided by the nature of the underlying system, these conditions need not only be deterministic but they can very well be stochastic, as well. Since starting and stopping conditions are part of the terminating simulations, relatively straightforward techniques can be used to estimate the parameters of interest in these experiments (e.g., method of independent replications.)

A *nonterminating simulation*, on the other hand, is the one for which there is no natural starting and stopping conditions. And the aim of a nonterminating simulation is to estimate the parameter of a steady-state distribution. An important characteristic of nonterminating simulations is that steady-state parameter of interest does not depend on the initial conditions of the simulation. However, steady-state exists only in the limit, that is, as the run length goes to infinity. And the run length of any simulation needs to be finite. Therefore, the initial conditions, which normally may not represent the system conditions in the steady-state, will apparently bias the estimates based on the simulation. This is called the *problem of initial transient* in the simulation literature. Many techniques have been proposed in the literature to remedy this problem (see, for example, Kelton, 1989; Kelton and Law, 1983; Schruben, 1981; Schruben et al., 1983; Goldsman et al., 1994; Vassilacopoulos, 1989; Welch, 1982; and White, 1997; among many others).

The initial transient problem deserves particular attention in any successful simulation study. Almost all of the studies in the literature are in the form of either method developments that try to mitigate the effects of initialization bias or works that compare the effectiveness of proposed techniques via applying them to analytically tractable models. That is, the studies done so far try to assess and improve the efficiency and efficacy of the earlier proposed techniques. We have never seen a study explicitly investigating how the initial transient period behaves with respect to different system conditions. We, in this thesis, are primarily interested in the *behavior* of the initial transient. More specifically, we are trying to observe the change in the length of the transient period of nonterminating simulations with changes in the system parameters.

Motivation of the proposed study

The primary motivation for this study comes from the often negligence of initial transient problem in practice, especially when using method of independent replications, and, more importantly, from non-existence of objective procedures to deal with this problem that are guaranteed to work well in every situation. In practice, most practitioners and even academic researchers, as well, often neglect the bias induced by the initial transient period in doing their simulation studies. The analysts, in general, truncate some initial portion of the whole sequence to mitigate the effects of initialization bias, however, they do this truncation in a rather informal way.

Furthermore, in comparing several alternative system designs, the truncation point is chosen by observing only one particular design, which may have a relatively short transient period, and the same amount of data is truncated from all other designs. This strategy, apparently, will not mitigate the effects of bias induced by initial conditions, if some designs have longer transient periods. In making unbiased comparisons among alternative designs, deleting the same amount of data from each design makes sense. However, the length of the transient period, i.e., the number of data to be truncated, might change drastically from one design to the other. Hence, if the same amount of truncation is to be made, then this should be chosen by selecting the longest transient period.

Initialization bias induces more severe problems in the simulation results if the output analysis has to be done by the method of independent replications, which is often the case due to its simplicity. Moreover, there is no objective criterion for data truncation that works well in every situation, which makes the problem even harder. This is, perhaps, one of the main reasons for its negligence. If some guidelines can be given by the researchers about the behavior of the transient period with respect to different system parameters, then the problem discussed above about the comparison of several system designs would be minimized, if not completely eliminated.

For the time being, we restrict our attention only to manufacturing systems, particularly serial production lines and job-shop production systems. The reason for choosing serial production lines and job-shop systems is that they are the building blocks of most manufacturing systems, and one can observe the simplest form of interactions among system components, which than can be generalized to larger systems. Additionally, these systems are of economic importance as they are still the most widely used ones in practical manufacturing. The contribution of this study to the literature can be stated as follows:

- It provides an extensive review of the literature on initial transient problem, which might be a starting point for future research in this area.
- It applies a relatively new truncation technique in addition to a frequently used visual tool, which allows us to assess its applicability in real-world system simulations, to discuss its theoretical limitations, and to give guidelines for its implementation.
- By giving detailed results for manufacturing systems, it provides a framework for simulation practitioners to validate their model findings regarding the transient period, a problem which has not received enough attention.

The organization of the thesis is as follows: after a short introduction to the initial transient problem in Chapter 1, it continues with presenting basic statistical results for the analysis of simulation outputs in Chapter 2. It then continues with a more precise definition of the problem of initial transient, which is followed by a literature survey. After giving some of the solution techniques to the problem of initial transient the chapter ends with a summary. We present the proposed study and the methodology used for this purpose in Chapter 3. Also, the methodology is illustrated by a detailed example. This chapter also discusses the system considerations and experimental parameters that are used in this study. Chapters 4 and 5 present the results of experiments for serial and job-shop production systems, respectively. This thesis ends with a conclusion chapter in which the results of the previous sections are summarized and future research directions are elaborated.

2 THE PROBLEM OF INITIAL TRANSIENT

2.1 Introduction

In this chapter, we study the problem of initial transient in more detail. The inherent variability in simulation data necessitates the use of statistical techniques to have meaningful conclusions from the simulation results. Unless appropriate statistical techniques are used to make the analysis, the results of a simulation experiment are always subject to suspect.

A classification of simulation studies based on the output analysis methods is made as either terminating or nonterminating (Law and Kelton, 2000; and Fishman, 1978). The starting and stopping conditions in a terminating simulation are in the nature of the system; hence the analyst has no control over it. Thus, by making several independent replications of the model one can use classical statistical techniques to analyze the output of terminating simulations. There is no natural starting and stopping conditions for the nonterminating simulations, hence both the way of starting and stopping do affect the performance measures to be estimated by these experiments. This problem should be remedied by careful statistical analysis instead of directly applying the classical methods. In the literature of initial transient problem, almost all the authors try to improve the efficiency and efficacy of the solution techniques proposed. We, in this study, primarily focus on the *behavior* of the *initial transient* (*warm-up* or *start-up*) period. We are interested in observing this behavior in the simulation of manufacturing systems, particularly in serial production lines and job-shop production systems. The reason for choosing these systems is that, they are the basic building blocks of more complex manufacturing systems. Additionally, the simplest forms of interaction among system entities can be observed easily, which than can aid to understand the behavior of larger systems.

One can skip Sections 2.2, 2.3, and 2.4 if s/he has enough statistical background and knowledge about the initial transient problem in simulation literature. Specifically, in Section 2.2 we present the basic results, such as calculation of the mean, variance, and confidence interval, from statistical theory. Then, we describe the behavior of a typical simulation output and define the initial transient problem in more precise terms in Sections 2.3 and 2.4, respectively. They are followed, in Section 2.5, by a presentation of the literature done so far on the initial transient problem. The chapter ends with presenting two of the data truncation techniques proposed in the literature in Section 2.6, which are used in this study.

2.2 Basic statistical concepts

Case 1. Independent and identically distributed sequences

Estimating a population mean from a sample data is a common objective of the statistical analysis of a simulation experiment. The mean is a measure of the central tendency of a stochastic process. The median is an alternative measure of the central tendency. In the cases when the process can take on very large and very small values, the median can be a better measure than the mean, since extreme values can greatly affect the mean. However, almost all the studies in the simulation literature deal with the mean rather than the median. Hence, in this study we will not use median, but mean instead.

By estimating the population mean we obtain a single number that serves as the best candidate for the unknown performance parameter. Although this single number, i.e., the point estimate, is important, it should always be accompanied by a measure of error. This is usually done by constructing a confidence interval for the performance measure of interest, which is helpful in assessing how well the sample mean represents the population mean. The first step in doing this is to estimate the variance of the point estimator. The variance is a measure of the dispersion of the random variable about its mean. The larger the variance, the more likely the random variable is to take values far from its mean.

Now, suppose that we have a stochastic sequence of n numbers, X_1, X_2,…, X_n. Further suppose that, this sequence is formed by independent and identically distributed random variables having finite mean μ and finite variance σ^2. Independence means that there is no correlation between any pairs in the sequence. Identically distributed means that all the numbers in the sequence come from the same distribution. Then, an unbiased point estimator for the population mean, μ, is the sample mean, $\overline{X}$, that is $E[\overline{X}] = \mu$, where

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{2.1}$$

An unbiased estimator of the population variance, σ^2, is the sample variance, S^2, that is $E[S^2] = \sigma^2$, where

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2 \tag{2.2}$$

Now, due to independence, the variance of the point estimator, $Var[\overline{X}]$, can be written as:

$$Var[\overline{X}] = \frac{S^2}{n} = \frac{1}{n(n-1)}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2. \tag{2.3}$$

The bigger the sample size n, the closer $\overline{X}$ should be to μ.

If n is sufficiently large, then an approximate 100(1-α) percent confidence interval for μ is given by

$$\overline{X} \pm z_{1-\alpha/2}\sqrt{Var[\overline{X}]} \tag{2.4}$$

Law and Kelton (2000) explain this confidence interval as follows: if a large number of independent 100(1-α) percent confidence intervals each based on n

observations are constructed, the proportion of these confidence intervals that contain (cover) μ should be 1-α. However, the confidence interval given by (2.4) holds only when the distribution of $\overline{X}$ can be approximated by a normal distribution, that is for sufficiently large n. If n is chosen too small, the actual coverage of the desired 100(1-α) percent confidence interval will generally be less than 1-α. To remedy this problem, z-distribution can be replaced by a t-distribution.

$$\overline{X} \pm t_{n-1,1-\alpha/2}\sqrt{Var[\overline{X}]} \tag{2.5}$$

Since $t_{n-1,1-\alpha/2} > z_{1-\alpha/2}$, the confidence interval given by (2.5) will be larger than the one given by (2.4) and will generally have coverage closer to the desired level 1-α (Law and Kelton, 2000).

Case 2. Correlated sequences

It is important to notice that the results presented so far are valid only if the assumptions of independence and identical distribution holds. If any of these assumptions is violated, then the results might change drastically. Now, we will examine the results when the assumption of independence is violated. We start with a definition of the covariance and correlation.

Let X and Y be two random variables, and let $\mu_X = E[X]$, $\mu_Y = E[Y]$, $\sigma^2_X = Var[X]$, and $\sigma^2_Y = Var[Y]$. The covariance and correlation are measures of the linear dependence between X and Y. The covariance between X and Y is defined as (Fishman, 1973b);

$$Cov[X, Y] = \mathrm{E}[(X- \mu_X)\cdot(Y- \mu_Y)] = \mathrm{E}[X\cdot Y] - \mu_X \cdot \mu_Y \tag{2.6}$$

The covariance can take on values between -∞ and ∞. The correlation coefficient, ρ, standardizes the covariance between –1 and 1.

$$\rho = \frac{Cov[X,Y]}{\sigma_X \sigma_Y} \tag{2.7}$$

If ρ is close to +1, then X and Y are highly positively correlated. On the other hand, if ρ is close to –1, then X and Y are highly negatively correlated. The closer ρ is to zero in both sides, the more independence between X and Y.

Now, suppose that we have a sequence of random variables X_1, X_2,… that are identically distributed but may be dependent. In such a time-series data, we can speak of *lag-j autocorrelation*.

$$\rho_j = \rho\ (X_i, X_{i+j}) \tag{2.8}$$

This means that the value of the autocorrelation depends only on the number of observations between X_i and X_{i+j}, not on the actual time values of X_i and X_{i+j}. If the X_i's are independent , then they are uncorrelated, and thus $\rho_j = 0$ for $j = 1, 2,\ldots$.

When we have an autocorrelated sequence X_1, X_2, …, X_n, the sample mean, $\overline{X}$, given by (2.1) is still an unbiased estimator of the population mean, μ. However, the estimation of the variance of the point estimate becomes a hard job (Banks et al., 1996);

$$Var[\overline{X}] = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} Cov(X_i, X_j) \tag{2.9}$$

Obtaining the above estimate is certainly a hard job since each term $Cov(X_i, X_j)$ may be different, in general. Fortunately, systems that have a steady-state will produce an output process that is approximately covariance-stationary (Law and Kelton, 2000). If a time-series is covariance-stationary, then the statement given in (2.9) can be simplified to (Moran 1959; Anderson, 1971):

$$Var[\overline{X}] = \frac{S^2}{n}\left[1 + 2\sum_{j=1}^{n-1}(1 - j/n)\rho_j\right] \tag{2.10}$$

It can be shown (see, Law, 1977) that the expected value of the variance estimator, S^2/n, is:

$$E\left[\frac{S^2}{n}\right] = B \cdot Var[\overline{X}] \tag{2.11}$$

where,

$$B = \frac{n/c - 1}{n - 1} \tag{2.12}$$

and c is the quantity in brackets in (2.10). Now,

(i) If X_i's are independent , then $\rho_j = 0$ for $j = 1, 2,\ldots$. Hence, $c = 1$ and (2.10) simply reduces to the familiar expression, S^2/n. Note also that $B = 1$, hence S^2/n is an unbiased estimator of $Var[\overline{X}]$.

(ii) If X_i's are positively correlated, then $\rho_j > 0$ for $j = 1, 2,\ldots$. Hence $c > 1$, which results that $n/c < n$, and $B < 1$. Therefore S^2/n is biased low as an estimator of $Var[\bar{X}]$. If this correlation term were ignored, the nominal 100(1-α) percent confidence interval would be misleadingly too short and its true coverage would be less then 1-α.

(iii) If X_i's are negatively correlated, then $\rho_j < 0$ for $j = 1, 2,\ldots$. Hence $0 \le c < 1$, which results that $n/c > n$, and $B > 1$. Therefore S^2/n is biased high as an estimator of $Var[\bar{X}]$. In this case, the nominal 100(1-α) percent confidence interval would have true coverage greater than 1-α. This is a less serious problem then the one in case (*ii*).

Conway (1963) gave an upper bound on the variance estimator for the correlated case, assuming that the correlation decreases geometrically with distance. That is, if the correlation between adjacent measurements is ρ, the correlation between non-adjacent measurements separated by one is ρ^2, separated by 2 is ρ^3, etc. The upper bound on this variance is given by:

$$Var[\bar{X}] < \frac{S^2}{n}\left[1 + \frac{2\rho}{1-\rho}\right] \qquad (2.13)$$

In queuing systems the autocorrelations are positive, i.e., if customer i has to wait relatively long then the next customer (i+1) probably has to wait long, too. The effect of positive autocorrelations was discussed above. Kleijnen (1984) states that the autocorrelations of M/M/1 systems might be so high that, the quantity in brackets in (2.10), c, would be as large as 360 when the traffic intensity is 0.90 and 10 when the traffic intensity is as low as 0.50.

To make the discussion complete, we make the definition of a covariance-stationary process. A discrete-time stochastic process $X_1, X_2,\ldots$ is said to be covariance-stationary if

$$\mu_i = \mu \text{ for } i=1,2,\ldots \text{ and } -\infty < \mu < \infty$$

$$\sigma_i^2 = \sigma^2 \text{ for } i=1,2,\ldots \text{ and } \sigma^2 < \infty$$

$$cov(X_i, X_j) \text{ is independent of } i \text{ for } j = 1, 2,\ldots$$

Thus, for a covariance-stationary process the mean and the variance are stationary over time (common mean and variance), and the covariance between two observations X_i and X_{i+j} depends only on the *lag j* and not on the actual time values of *i* and *i+j*.

2.3 The behavior of simulation outputs and analysis methods

Simulations of dynamic, stochastic systems can be classified as either terminating or nonterminating depending on the criterion used for determining the run length. The terms transient-state and steady-state are also used extensively for terminating and nonterminating simulations, respectively.

In a terminating simulation, the starting and stopping conditions of the model are explicitly dictated by the system to be modeled and the analyst has no control over it. That is, the starting and stopping conditions used in a terminating simulation is determined by the “nature” of the system to be modeled. Hence, a transient-state simulation is needed if the performance of a stochastic system under some pre-determined initial and terminating conditions is of interest. This means that the measures of performance for a terminating simulation depends explicitly on the state of the system at time zero. Therefore, special care should be given in choosing the initial conditions for the simulation at time zero in order those conditions to be representative of the initial conditions for the corresponding system. In a terminating system, a possible transient behavior forms part of the response (Kleijnen, 1975). Although determined explicitly by the nature of the system, the initial and terminating conditions need not be deterministic; rather, they can very well be of stochastic nature, as well. The only way to obtain a more precise estimate of the desired measures of performance for a terminating simulation is to make independent replications of the simulation. This will produce estimates of the performance measure of interest that are independent and identically distributed. Hence the formulas presented under “*Case* 1” heading of Section 2.2 are readily applicable to have a point estimator for the population mean of the process and to assess the precision of this point estimator. However, note that confidence intervals are approximate because certainly they depend on the assumption that estimates are normally distributed, which is merely satisfied in practice. If enough replications are done, the output analysis for a terminating simulation becomes fairly simple since the classical methods of statistical analysis can be directly applied. However, this is not the case for a steady-state (or non-terminating) simulation.

In a nonterminating simulation, however, there is no explicitly defined way of starting and stopping the model. A steady-state simulation is applicable if the system's performance, which is independent of any initial and terminating conditions, is to be evaluated. In other words, the desired measure of performance for the model is defined as a limit as the length of the simulation goes to infinity. The usual assumption is that a nonterminating simulation achieves a stochastic steady-state in which the distribution of the output is stationary in some sense (typically in the mean) and ergodic (independent of any specific initial or final run conditions) (Law, 1984).

Note that stochastic processes for most real-world systems do not have steady-state distributions, since the characteristics of the system change over time. On the other hand, a simulation model may have steady-state distributions, since the characteristics of the model are often assumed not to change over time. It is also important to note that a simulation for a particular system might be either terminating or nonterminating, depending on the objectives of the study. Law (1980) contrasts these two types of simulations in several examples, and discusses the appropriate framework for statistical analysis of the terminating type (i.e., method of independent replications).

Let X_1, X_2,… be an output process from a single simulation run. The X_i's are random variables that will, in general, be neither independent nor identically distributed. The data are not independent, because there will, in general, be a significant amount of correlation among observations, i.e., autocorrelated sequence. For example, in the simulation of a simple queuing system, if the j^{th} customer to arrive waits in line for a long amount of time, then it is quite likely that the $(j+1)^{st}$ customer will also wait in line for a long amount of time, and vice versa. Additionally, they are not identically distributed, either because of the nature of the simulation model or of the initial conditions selected to start the simulation. Not identically distributed means that the sequence is nonstationary, that is the distributions of the output observations change over time, i.e., $E[X_i] \neq E[X_{i+1}]$. However, for some simulations X_{d+1}, X_{d+2},… will be approximately covariance-stationary if d is large enough, where the number d is the length of the warm-up period (Nelson, 1992). For example, the first observation on the process of interest is a function of the initial conditions, because of the inherent dependence among events. The second observation is also a function of the initial values but usually to a lesser extent than the first observation is. Successive observations are usually less dependent on the initial conditions so that eventually events in the simulation experiment are independent of them. If the initial conditions are not chosen from the steady-state behavior of the output process, then the rate of this dependence scheme will hardly diminish compared to the rate when appropriate initial conditions are chosen. For these two reasons, classical statistical techniques, which are based on independent and identically distributed data are not directly applicable to the outputs of steady-state simulation experiments. Note that the problem of dependence can be solved by simply making several independent replications of the entire simulation. However, the problem with not identically distributed (nonstationary) data can not be solved by independent replications. It needs further careful analysis, which is, in fact, a major part of this thesis.

Several works have been done on the statistical analysis of steady-state simulations. These techniques can be classified under one of the following six methods (Law and Kelton, 2000), for which we also give short descriptions; replication, batch means, autoregressive method, spectrum analysis, regenerative method, and standardized time-series modeling. A detailed survey about the first five methods can be found in Law and Kelton (1982, 1984). (Note that due to the main focus of this study, we give the method of independent replications in more detail in a separate heading.)

- *Method of batch means*. This method is based on a single long run and seeks to obtain independent observations. However, since it is based on a single run, it has to go through the transient period only once. Transient period is the period in a simulation run that starts with initialization and continues until the output sequence reaches a stationary distribution. More precise definition will be given in the next section. The output sequence X_1, X_2, …X_k (assuming that the initial transient is removed) is divided into n batches of length m (note that, $k = m \cdot n$). Letting $\overline{X}_j = (1/m)\sum_{i=1}^{m} X_i$, for j = 1, 2,…, n, be the sample mean of the m observations in the j^{th} batch, we obtain the grand sample mean as

$\overline{\overline{X}} = (1/n)\sum_{j=1}^{n} \overline{X}_j$. Hence, $\overline{\overline{X}}$ serves as the point estimator for the true performance measure. If the batch size, *m*, is chosen sufficiently large, then it can be shown that $\overline{X}_j$'s will be approximately uncorrelated (Law and Carson, 1979). Then classical statistical techniques can be applied. If the batch size is not chosen large enough, then $\overline{X}_j$'s will be highly correlated. The effect of negligence of correlation in a sequence was discussed in the "*Case* 2" heading of Section 2.2.

- *Autoregressive method*. This method was developed by Fishman (1971, 1973b, and 1978). It tries to identify the autocorrelation structure of the output sequence and then makes use of this structure to estimate the performance measures. A major concern in using this approach is whether the autoregressive model provides a good representation of the stochastic process.
- *Spectrum analysis*. This method is very similar to the autoregressive method. Likewise, it also tries to use the estimates of the autocorrelation structure of the underlying stochastic process to obtain an estimate of the variance of the point estimator. However, it is, perhaps, the most complicated method of all, requiring a fairly sophisticated background on the part of the analyst.
- *The regenerative method*. This method was simultaneously developed by Crane and Iglehart (1974a, 1974b, and 1975) and Fishman (1973a). The idea is to identify random times at which the process probabilistically starts over, i.e., regenerates, and use these regeneration points to obtain independent and identically distributed random variables to which classical statistical analysis can be applied to estimate the point estimator and its precision. In order for this method to work well, there should be short but a large number of cycles, where a cycle is defined as the interval between two regeneration points. The difficulty with using regenerative method in practice is that real-world systems may not have easily identifiable regeneration points, or even if they do have, expected cycle length may be so large that only a few cycles can be simulated. A fairly good discussion is given in Crane and Lemoine (1977).
- *The standardized time-series method*. This method is based on the same underlying theory as Schruben's test (1982). It assumes that the process X_1, X_2,…

is strictly stationary with μ for all i and is also phi-mixing. Strictly stationary means that the joint distribution of X_{1+j}, X_{2+j},…,X_{n+j} is independent of j for all time indices 1, 2,…,n, where j is the analyst specified batch size. Also, X_1, X_2,… is phi-mixing if X_i and X_{i+j} become essentially independent as j becomes large. The major source of error for this method is choosing the batch size, j, small.

It should be noted that one important common characteristic of all the methods given above is that they are based on only a single long run of the simulation model. Standardized time-series may be applied to either a single long run or multiple short runs. This is an important difference between these methods and the method of independent replications, which we discuss in the next subsection. The comparison of having one long run versus having multiple independent short runs is made by Whitt (1991). He provides examples showing that each strategy can be much more efficient than the other, thus demonstrating that a simple unqualified conclusion is inappropriate. He also adds that doing fewer runs (e.g., only one) is more efficient when the autocorrelations decrease rapidly compared to the rate the process approaches steady-state. Furthermore, the method of batch means, the autoregressive approach, and the spectrum analysis assume that the process X_1, X_2,… is covariance-stationary, which will rarely be true in practice. However, this problem can be remedied by choosing a sufficiently large d, and keeping only the sequence after d, i.e., X_{d+1}, X_{d+2},… .Additionally, all the methods, including the method of independent replications suffer, from the initial transient problem except the regenerative method (Kleijnen, 1984).

The most serious problem in the method of independent replications is the bias in the point estimator, whereas the most serious problem in the other methods is the bias in the variance of the point estimator (Law and Kelton, 2000). They actually have a bias problem in the point estimator, however, since the run length is relatively very large when compared to the run length of the method of replications, this problem diminishes out. Next, we discuss the method of independent replications in more detail.

Method of independent replications

The motivation for this study was in the method of independent replications. Here, several independent and identically operated simulations are made in order to produce independent and identically distributed observations $\overline{X}_1, \overline{X}_2, ...,$ where $\overline{X}_j$ is the output measure of interest from the j^{th} entire simulation run. In outline, this method is identical to that used for terminating simulations (Law, 1980). Independence is satisfied by using a separate random number stream for each replication. And, the assumption of identical distribution is satisfied by initializing and terminating the replications exactly in the same way.

A more precise explanation of the method is as follows. Suppose we make n independent replications of a simulation model each of length m.

Table 2.1 Typical simulation output using method of independent replications.

Simulation			*Observations*						
Replications	O_1	O_2	…	O_d	O_{d+1}	O_{d+2}	…	O_m	$\overline{X}_j$
R_1	X_{11}	X_{21}	…	X_{d1}	$X_{d+1,1}$	$X_{d+2,1}$	…	X_{m1}	$\overline{X}_1$
R_2	X_{12}	X_{22}	…	X_{d2}	$X_{d+1,2}$	$X_{d+2,2}$	…	X_{m2}	$\overline{X}_2$
.	.				.			.	.
.	.				.			.	.
.	.				.			.	.
R_n	X_{1n}	X_{2n}	…	X_{dn}	$X_{d+1,n}$	$X_{d+2,n}$	…	X_{mn}	$\overline{X}_n$
									$\overline{\overline{X}}$

Note that the observations within a single replication (row) are not independent. However, the observations in a single column are apparently independent. Additionally, the expectation of the sample mean of each replication (i.e., $\overline{X}_j$'s) is not equal to the population mean, μ. That is, the observations within a single run are not identically distributed, as well. The reason for this biasing effect is that the initial conditions for the replications cannot be specified to be representative of the steady-state conditions. Each run, therefore, will involve the same problem of achieving steady-state as the first.

For the time being, assume that the observations within a replication come from an identical distribution. Given this assumption, and the observations obtained from each replication, since each row shows an independent replication of the simulation model, we obtain point estimates, $\overline{X}_j$ for j = 1, 2,…,n, of the population mean, μ, where $\overline{X}_j$ is given by:

$$\overline{X}_j = \frac{1}{m}\sum_{i=1}^{m} X_{ij}, \quad \text{for } j = 1, 2, ... n \tag{2.14}$$

and, X_{ij} is the i^{th} observation in the j^{th} replication. Thus, the grand mean, $\overline{\overline{X}}$, is the ultimate point estimator for μ and is given by:

$$\overline{\overline{X}} = \frac{1}{n}\sum_{j=1}^{n} \overline{X}_j \tag{2.15}$$

The variance of the point estimator and the confidence interval for the mean can be estimated by equations given in (2.3) and (2.5), respectively.

The method of independent replications has two principal advantages (Kelton, 1989):

(i) It is a very simple method to apply, by far the simplest of the six techniques mentioned above. Also, most simulation languages have some facility for specifying that a model be replicated some number of times,

(ii) It produces the independent and identically distributed sequence $\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n$ of observations on the system, to which the methods of classical statistics may be directly applied. This includes not only the familiar confidence interval construction and hypothesis testing, but also methods such as multiple ranking and selection procedures, a very useful class of techniques in simulation since several alternative system designs are frequently of interest.

Now, if the assumption of identically distributed data is relaxed the situation becomes complicated. Identically distributed data can not be obtained from a single simulation run, because the initialization methods used to begin most steady-state simulations are typically far from being representative of the actual steady-state conditions. This leads to bias in the simulation output, at least for some early portion of the run. It should be noted that increasing the number of replications will not cure this problem, rather each replication will be effected by the initialization bias problem. This is a major problem in the method of independent replications.

2.4 Problem definition

The problem of initial transient was tried to be explained in previous sections, but rather in an informal and inconvenient manner. Here, we will define the problem of initial transient in a mathematically more precise way and also discuss the effects of this problem on the simulation results. In order for the section to be complete in itself we first draw the borders of our environment.

Two types of simulation studies that should be distinguished are terminating (transient-state) and nonterminating (steady-state) simulations, when the output analysis methods are of concern. Output analysis deals with the estimation of some performance measures of interest. The analyses of the outputs for terminating simulations can easily be done by employing classical statistics, if caution was taken in determining independent and identically distributed outcomes. By simply replicating the entire simulation run with different random number streams, this goal can be achieved. However, this job, i.e., estimating performance measures, is not as easy for steady-state simulations as it is for terminating ones. Several techniques have been proposed in the literature to obtain accurate and precise steady-state performance measures, where we solely concentrate our study on the method of independent replications.

Two curses of steady-state discrete-event simulation are the autocorrelations and the initial bias. Autocorrelations occur because new states of the process typically depend strongly on previous ones. The existence of autocorrelations in a sequence produces bias in the estimate of the variance of the point estimator. This problem can be remedied by simply having multiple independent runs of the model. Since the output sequence of a single run exhibits autocorrelated behavior, the initially specified conditions can and will greatly affect the outputs obtained. Bias is the difference between the expected value of the point estimator, here presumed to be the sample mean, and the true value of the quantity to be estimated, μ. Initialization bias is the bias occurring due to initializing the simulation model in a condition that is not representative of the steady-state conditions.

The biasing effect of initial conditions can not be mitigated by having multiple independent runs, because each run is initialized in the same way as the others and this bias, if exists, occurs in all the runs.

A formal definition of the problem of initial transient can now be given as follows:

Suppose that $X_1, X_2, \ldots, X_m$ is the output process from a single run of the simulation for which a set of initial conditions, denoted by I_0, exist at $i = 0$. Also suppose that these random variables have a steady-state distribution, denoted by F, which is independent of the initial conditions I_0, with the first moments given as follows:

$$\lim_{i \to \infty} E[X_i \mid I_0] = \lim_{i \to \infty} E[X_i] = \mu \tag{2.16}$$

where μ is the steady-state mean value. The goal of a steady-state simulation of the process $X_1, X_2, \ldots$ is to estimate this mean μ and construct a confidence interval for this mean. Note that, in practice, any simulation run will necessarily be finite. Therefore, the initial conditions will clearly affect the point estimators. Also, since it will generally not be possible to choose the initial conditions for the simulation to be representative of "steady-state behavior," the distribution of the X_i 's (for $i = 1, 2, \ldots$) will differ from F over time. Furthermore, an estimator of μ based on the observations $X_1, X_2, \ldots, X_m$ will not be "representative." For example, the sample mean, $\overline{X}$ that is given by (2.14), will be a biased estimator of μ for all finite values of m. The problem that has just been described is called *the problem of initial transient* or *the start-up* (or *warm-up* or *initialization bias*) *problem* in the simulation literature.

Once this problem is recognized by the analyst, it should be given enough effort to mitigate its effects. There is an agreement in the literature on using the mean-squared error (MSE) of the point estimator to assess the efficacy of any proposed remedial procedure to this problem. We defer a rigorous definition of the MSE until Section 2.6 and content with stating that it is the sum of the variance of the point estimator and the square of the bias in the point estimator. The smaller the MSE, the better the output sequence is.

Mitigating the effects of initialization bias

Some authors suggest, for special systems, that retaining the whole sequence and estimating the performance measures of interest with this whole sequence would minimize the MSE (Kleijnen, 1984). Indeed, Law (1984) proved that for simple queuing systems MSE is minimized by using the whole series, assuming that the system started in the empty and idle state and the length of the run is long. However, even if the MSE would be minimal, the resulting confidence interval may be inconsistent. If significant bias remains in the point estimator and a large number of replications are used to reduce the point estimator variability, then a narrow confidence interval will be obtained but around the wrong quantity (Adlakha and Fishman, 1982; Law, 1984). This happens, because bias is not affected by the number of replications; it is affected only by deleting more data or extending the length of each run. In the presence of initial-condition bias and a tight budget, the number of replications should be small and the length of each replication should be long (Nelson, 1992).

Conway (1963) suggests using a common set of starting conditions for all systems, when two or more alternative systems are compared.

The length of the transient period will certainly depend on the method used for initialization. Having this fact in mind, one can suggest to start the simulation in a state, which is “representative” of the steady-state distribution. This method is sometimes called *intelligent initialization* (Banks et al., 1996). This approach can be implemented in two ways. The first is called *deterministic initialization*, where the initial conditions are chosen as constant values such as the mean or the mode of the steady-state behavior of the process. A second way, called *stochastic initialization*, tries to estimate the steady-state probability distribution of the process and then uses this estimated distribution to draw the initial conditions instead of specifying it to be the same deterministic value for each replication. Estimating this distribution can be done either by having pilot runs or by using the results of similar systems that can be solved analytically. The replications in stochastic initialization, though actually begin in generally different numerical positions, are still independent and identically distributed since the rule by which the initial states are chosen is always the same.

Indeed, the ideal in intelligent initialization techniques would be to draw the initial state from the steady-state distribution itself. But an analyst in possession of this information would have no reason to execute a simulation. Even if this is done satisfactorily, this can only decrease, but not completely eliminate, the time required to for the simulation to achieve steady-state. There will still be an initial period during which the expected system state will differ from the desired steady-state expectation, but hopefully, both the duration of the transient period and the magnitude of the bias will be diminished.

Another, yet more practical and most often suggested technique in the literature for dealing with the problem of initial transient is known as the initial data *deletion* (or *truncation*) or *warming-up* the model. The idea is to delete some number of observations from the beginning of a run and to use only the remaining observations to estimate the steady-state quantities of interest. Since our thesis mainly concentrates around this technique we devote a separate section for data truncation techniques and discuss the detailed mechanics in Section 2.6.

2.5 Literature survey

The literature in the initial transient problem can be divided into two broad categories; studies centered around intelligent initialization and studies centered around truncation heuristics. Truncation heuristics, indeed, can also be classified as either heuristics that suggest a truncation point or recursive applications of hypothesis testing to detect the existence of initialization bias. Furthermore, some authors have studied the assessment of proposed techniques both in theoretical limitations and in practical applicability. Here, we review the important studies that have taken considerable attention in the literature.

The problem of initial transient has challenged the researchers so much that even PhD dissertations solely devoted to this problem have been done (see, for example, Morisaku, 1976; Murray, 1988). Table 2.2 summarizes the literature on the initial transient problem.

Table 2.2 Summary of the literature on the initial transient problem.

Type of study	Studies conducted
Intelligent initialization	
Deterministic initialization	Madansky, 1976; Kelton and Law, 1985; Kelton, 1985; Murray and Kelton 1988a; Blomqvist, 1970
Stochastic initialization	Kelton, 1989; Murray, 1988; Murray and Kelton, 1988b
Antithetic initial conditions	Deligönül, 1987
Truncation heuristics	
Graphical techniques	Welch, 1981, 1982, 1983
Repetitive hypothesis testing	Schruben 1982, Schruben et al., 1983; Schruben and Goldsman, 1985; Goldsman et al., 1994; Schruben, 1981; Vassilacopoulos, 1989
Analytical techniques	Kelton and Law, 1983; Asmussen et al., 1992; Gallagher et al., 1996; White, 1997, White et al., 2000; Spratt, 1998
Surveys	Gafarian et al., 1978; Wilson and Pritsker, 1978a, 1978b; Chance, 1993
Assessments	Conway, 1963; Gafarian et al. ?; Law, 1975, 1977, 1984; Kelton, 1980; Kelton and Law, 1981, 1985; Fishman, 1972, 1973a; Adlakha and Fishman, 1982; Kleijnen, 1984; Cash et al., 1992; Nelson, 1992; Ma and Kochhar, 1993; Snell and Schruben, 1985;
Others	Glynn and Heidelberger, 1991, 1992a, 1992b; Heidelberger and Welch, 1983; Nelson, 1990

The first study that deals with the initial bias in the simulation output data is due to Conway (1963). Although the problem of initial transient was recognized so early, and many efforts was given to solve the problem, there still does not exist a general objective rule or procedure as a solution to the problem. Conway (1963) also proposed a method, which is perhaps the first formal truncation heuristic. Applying this rule, the output sequence is scanned using a forward pass, beginning with the initial condition, to determine the earliest observation (in simulated time), which is neither the maximum nor the minimum of all later observations. This observation is taken as the truncation point for the current run.

Several other methods have been developed in the literature. Gafarian et al. (1978) found that none of the methods available at that time performed well in practice and they stated that none of them should be recommended to practitioners. They also proposed an alternative heuristic, similar to the Conway's rule. Applying this alternative, the output sequence is scanned using a backward pass, beginning with the last observation, to find the earliest observation (in simulated time) that is neither the maximum nor the minimum of all earlier observations. This observation is taken as the truncation point for the current run. Gafarian et al. (?), in another study, suggest the following criteria to assess the efficiency and effectiveness of a proposed technique; accuracy, precision, generality, simplicity, and cost. However, all these criteria are subjective and hard to accurately estimate in practice.

Wilson and Pritsker (1978a) also surveyed the various simulation truncation techniques. They concluded that the truncation rules of thumb are very sensitive to parameter misspecification, and their use can result in excessive truncation. Wilson and Pritsker (1978b), in another study, evaluated finite state space Markov processes and found that choosing an initial state near the mode of the steady-state distribution produces favorable results. However, in practice that mode is unknown. Anyhow, these results suggest that the empty state is not the best starting point for data collection if runs are replicated. They also noted that it is more effective to choose good initial conditions than to allow for long warm-up periods.

Chance (1993) also provides a survey of the works done in the initial transient problem in simulation literature, which is fairly recent when compared to the above surveys.

Perhaps the simplest and most general technique for determining a truncation point is a graphical procedure due to Welch (1981, 1982, and 1983). In general, it is very difficult to determine the truncation point from a single replication due to the inherent variability of the process $X_1, X_2, \ldots$. As a result Welch's procedure is based on making n independent replications of the simulation and averaging across replications. Further reduction in the variability of the plot is achieved by applying a moving average. The moving average window size, w, and the number of replications, n, are increased until the steady-state stabilization point is obvious to the practitioner. Law and Kelton (2000) recommended Welch's plotting technique, with its subjective assessment, as the simplest and most general approach to detect the completion of the transient phase. One drawback of Welch's procedure is that it might require a large number of replications to make the plot of the moving averages reasonably stable if the process itself is highly variable.

Kelton (1989) suggested estimating the steady-state distribution from a "pilot" run, and then independently sampled from this estimated distribution in order to determine the initial conditions for each production run, which is also known as random initialization. He used the maximum entropy rule proposed by Jaynes (1957). He found that random initialization reduces the severity and duration of the initial transient period as compared to starting the simulation in a fixed deterministic state. He also reported that bias reduction comes at no substantial increase in the variance of the point estimator, and the mean-squared error is often reduced; confidence intervals exhibit improved coverage probabilities without significant increase in half-length. It is recommended that, for relatively short runs in the context of the method of independent replications, steady-state simulations are initialized stochastically rather than deterministically. However, this technique would be harder to apply in the case of many real-world simulations (Murray, 1988). A similar approach was followed by Deligonul (1987), with the exception of starting with antithetic conditions rather than random conditions.

Murray and Kelton (1988b), in an accompanying study, described the results of an analytical study on the effectiveness of random initialization. They used a first-order autoregressive process, and showed that random initialization is effective in reducing bias in the point estimate and increasing coverage of the interval estimate without unduly increasing variance or mean square error.

Schruben (1982) developed a very general procedure based on standardized time-series for determining whether the observations X_{d+1}, X_{d+2}, …, X_m (d not need be zero) contain initialization bias with respect to the steady-state mean, μ. However, this procedure is not an algorithm for determining a deletion amount d, but rather a test to determine whether a set of observations contains initialization bias. It could be applied to the observations remaining after some amount of deletion has been done on a set of output data, in order to determine if there is remaining bias. Another way of implementing this method can be recursively deleting some amount of data and checking for initialization bias until the test concludes that no bias is left in the sequence. However, this might be a too time consuming task. This study has been the building block of several other studies (Schruben and Goldsman, 1985; Schruben et al., 1983). One restriction of this test is that it is applicable to only univariate output. The theoretical framework for the multivariate case is also given by Schruben (1981).

Goldsman et al. (1994) present a family of tests to detect the presence of a transient mean in a simulation output, which are natural generalizations of Schruben's work. The tests, namely, the batch means test (BM), area test (AREA), maximum test (MAX), combined BM+AREA test, and combined BM+MAX test, compare the variance estimators from different parts of a simulation output, and are based on the method of batch means and standardized time-series. They also provide a power analysis of the tests. Roughly, tests work as follows: The output process $X_1, X_2, \ldots, X_m$ is partitioned into two contiguous, nonoverlapping portions. An estimate of the variance of the sample mean is calculated based solely on the first portion of the output, and then based solely on the latter portion of the output. A large difference between these two estimates is unlikely if the process is stationary. Otherwise, the null hypothesis of no initial bias is rejected.

Cash et al. (1992) tested some initial condition bias detection tests on analytically tractable models such as first-order autoregressive model, M/M/1 queue, and a Markov chain model. The tests under consideration were the ones developed by Goldsman et al. (1994). They reported that the tests were powerful in detecting bias when the bias is severe at the very beginning of the output sequence, but dies out quickly. However, if the bias decays slowly, it became harder for the tests to detect the bias. The MAX test was found to be the most powerful.

Vassilacopoulus (1989) proposed a hypothesis test based truncation point detection procedure. It has the advantage of not requiring the estimation of the variance of a given stochastic sequence.

Ma and Kochhar (1993) presented a comparison study of two initial bias detection tests, namely the optimal test of Schruben (1982) and the rank test of Vassilacopoulus (1989), using output sequences with known transient functions. Their results showed that both tests perform satisfactorily in a similar way and appear to be powerful and efficient, although the optimal test tends to be able to detect a few more biased sequence than the rank test, especially in the situations in which the initialization bias is less prominent. However, they recommended using rank test due to its ease of implementation and negligible difference between the performances of two tests.

White (1997) introduced a relatively new truncation heuristic named as Marginal Confidence Rule (MCR) and compared it with several alternative truncation heuristics proposed earlier. Simply stated, this rule minimizes the width of marginal (within run) confidence interval about the sample mean of the reserved observations. His results showed that MCR dominated other rules while all were effective in improving the accuracy of the point estimator without undue loss of precision. Ease of understanding and implementation, inexpensive computation, efficiency in preserving representative simulation data, and effectiveness in mitigating the initial bias were stated as the advantages of the new rule.

White et al. (2000) further elaborated on the MCR rule and renamed it as the Marginal Standard Error Rule (MSER) with almost no modification. They tested two variants of the MSER rule and three variants of Schruben's test in mitigating the effects of initialization bias using a second-order autoregressive process with known bias function. Results confirmed that four of the five rules were effective and reliable, consistently yielding truncated sequences with reduced bias. In particular, the MSER heuristics outperformed the BM, MAX tests presented in Goldsman et al. (1994) and IE test presented in Nelson (1992), with Spratt's (1998) MSER-5 being the most effective and robust choice for general-purpose method. Additionally, BM test was found to be the least effective and the other methods generally fell somewhere between that of BM and MSER-5. Another advantage of MSER-5 rule was stated as that it requires least amount of computation time.

Madansky (1976) considered simulation of an M/M/1 queue with any number of customers present at time zero. He showed that, for large m (run length), initializing the system empty and idle minimizes the mean-squared error of the point estimate. He notes that empty state is the mode of the steady-state distribution of the number in system. He also concludes that increasing the run length is more advantageous than replication in terms of mean-squared error.

Kelton and Law (1985), Kelton (1985), and Murray and Kelton (1988a) found that, for $M/M/s$, $M/E_m/1$, $E_m/M/2$, and $M/E_m/2$ queues, initializing in a state at least as congested as the steady-state mean (as opposed to the mode) induced comparatively short transients. Optimal initial states were also found. In order to implement such ideas, some *a priori* estimate of the mode or the mean of the steady-state distribution would be required, perhaps from debugging or pilot runs of the model.

Kelton and Law (1983) developed an alternative algorithm for choosing *d* (the deletion point) and *m* (the run length) simultaneously that worked well for a wide variety of stochastic models based on linear regression. Their method entails generating the output sequence for each simulation replication in incremental blocks. After each output increment, the estimated mean for small batches is modeled as a population mean plus a zero-mean noise term. By fitting regressions on the means of the batches for the replications and testing the hypothesis of zero slope, the simulation run length and truncation point are estimated. However, a theoretical limitation of the procedure is that it basically makes the assumption that $E[X_i]$ is a monotone function of *i*. Additionally, a practical drawback of the algorithm is that it requires the analyst to set nine parameters. Moreover, they found that the steady-state mean appears to provide a better guide to initialization than does the mode. They also suggest to start in an undercongested state rather than in an "equally" overcongested state; i.e., "undershooting d^* is better than overshooting it by the same amount. In particular, empty and idle initialization is better than initializing with $2d^*$.

Fishman (1972) used a first-order autoregressive scheme to investigate the effects of initial conditions in a simulation on the estimation of the population mean. The effects are measured by bias and variance, that is by mean-squared error. The results show that elimination of observations near the beginning of the simulation reduces bias, as intended, but increases variance, sometimes significantly.

Fishman (1973a) presents the well-known regenerative method for estimating the sample performance measures in queuing simulations, for removing bias in the sample measures due to initial conditions.

Gallagher et al. (1996) provides an algorithm for determining the appropriate initial data truncation point for univariate output using a Bayesian technique called Multiple Model Adaptive Estimation (MMAE) with three Kalman filters. The estimated truncation is selected when the MMAE mean estimate is within a small tolerance of the assumed steady-state. The technique also entails averaging across independent replication.

Asmussen et al. (1992), in a very technical paper, studied analytical detection of stationarity in the initial transient problem. They particularly interested in regenerative processes and Markov processes with finite, countable, or general state space. Eight algorithms were presented for different cases, where the results were reported as both positive and negative. They also proved, in a mathematically precise sense, that without some restrictions on the class of simulations to be considered, there can exist no universally satisfactory means of detecting stationarity in a stochastic sequence.

Glynn and Heidelberger (1991) studies the theoretical problems in initial transient deletion in the multiple replicate steady-state simulation. They show that without any transient deletion, significant convergence problems can arise if the length of each replication is not considerably larger than the total number of replications. They further studied this problem in a theoretical and experimental framework for simulation replications executed on multiple processors in parallel in Glynn and Heidelberger (1992a and 1992b), respectively.

Heidelberger and Welch (1983) considered the problem of automatic generation of a confidence interval of prespecified width when there is an initial transient present in the output sequence. They gave a procedure by combining the Schruben's initial transient removal scheme and a run length control procedure of their own. The procedure is evaluated empirically for a variety of output sequences. The results show that, if the output sequence contains strong transient, then the procedure gives point estimates with lower bias, narrower confidence interval, and shorter run lengths when compared to the case of no check for the initial transient.

Variance reduction techniques, which is a very broad area in the simulation literature, in the presence of initial condition bias is studies by Nelson (1990).

2.6 Data truncation techniques to reduce initialization bias

In this section, we discuss data truncation techniques to remedy the problem of initial bias in general, and concentrate on two remedial approaches that are used in this thesis, namely, cumulative averages plot and Marginal Standard Error (MSER) heuristics.

Consider a discrete-event simulation that generates output sequences $\{X_{ij} : i = 1, 2,\ldots, m; j = 1, 2,\ldots,n\}$ with i being the observation number and j being the replication number. Each replication is started with the same set of initial conditions, I_0. Under the usual interpretation, the steady-state mean for the process

$$\mu = \lim_{i\to\infty} E\left[X_{ij} \middle| I_0\right] \qquad \forall\, j, I_0 \tag{2.17}$$

could be determined from any single run, regardless of the run number or initial conditions, if only the output sequence could be extended to include an infinite number of observations. Clearly, this is not practically possible.

The usual technique to deal with initialization bias, within the context of the method of independent replications, is *deletion* (or *truncation*). This is accomplished by deleting a portion of the output data from the beginning of each replication, and the deleted portion is assumed to account for the warming up of the system to steady-state conditions. Mathematically, d observations are truncated from the output sequence, and

$$\overline{X}_j(m,d) = \frac{1}{m-d}\sum_{i=d+1}^{m} X_{ij} \qquad \text{for } j = 1,2,\ldots,n \tag{2.18}$$

are used as a basic unit for analysis.

The objective is to remove observations that are rare and therefore atypical of individual observation sequences of a *fixed length*. The truncation assumption is that, given the arbitrary selection of initial conditions, these rare observations (if present) are most likely massed at the beginning of the output sequence. Hence, after the simulation has run for a sufficient "start-up" or "warm-up" period, the current observations should be more representative of steady-state than an arbitrary initial condition. Therefore, if we discard the observations collected during the warm-up period, in effect we are letting the behavior of the simulation choose a less biased initial condition for the reserved sequence. By truncating some data from the initial portion of the sequence, we are trying to improve the accuracy of the point estimator. However, extensive truncation would imply a loss of information and, more importantly, a loss of precision. Furthermore, it is *not* guaranteed to produce a better point estimate when truncation is applied to any particular sequence. Attention should be given to the character of the actual sequence deleted, relative to the actual sequence reserved, and not simply to the length or duration of the truncation sequence.

The practical difficulty with this deletion idea is determining values for d and m. For our purposes, we assume m to be fixed and deal only with determining d. If the sole purpose was to minimize the bias in a strict, mathematical programming sense, then clearly m should be as large as possible (ideally, $m \to \infty$) and d should be chosen as m-1 (Kelton, 1989). Such a formulation is not appropriate, since this solution would be unnecessarily wasteful of data. Instead, our problem is to find a value for d such that $E[\overline{X}_j(m,d)]$ is sufficiently near μ (if it exists) to allow us to treat the $\overline{X}_j(m,d)^{'}$s as being independent and identically distributed and unbiased for μ. If m, in addition to d, was also a parameter of interest to be determined by the analyst, then clearly several alternative $\{(m,d)\}$ pairs would satisfy this goal.

Some authors have questioned the efficacy of data deletion. Although deletion often decreases bias, in some cases it might also increase the variance of the estimator. The most commonly suggested measure of point estimator quality is the mean-squared error (Fishman, 1972), which is given by

$$\begin{aligned} MSE[\overline{X}_j(m,d)] &= Bias^2[\overline{X}_j(m,d)] + Var[\overline{X}_j(m,d)] \\ &= \{E[\overline{X}_j(m,d)] - \mu\}^2 + \{E[(\overline{X}_j(m,d) - E[\overline{X}_j(m,d)])^2]\} \end{aligned} \tag{2.19}$$

For the first-order autoregressive process, Fishman (1972) showed that variance of the point estimator increases while the bias in the estimator decreases. This was further studied by Snell and Schruben (1985), Kelton (1980), and Kelton and Law (1981). They showed that deletion may either increase or decrease the MSE of the point estimator, depending on m, d, and the values of the process parameters. They reported that deletion most significantly reduce the MSE when the initialization bias was high and the autocorrelation was heavy, causing the bias to dissipate slowly. In these cases, the value of d that minimized MSE decreased as m increased. Also, Blomqvist (1970) showed that for the M/M/1 queue and for certain other queuing systems with m sufficiently large, zero is that value of d which minimizes the MSE of $\overline{X}(m,d)$. However, MSE is a theoretical statistic that involve expectations taken over an infinite number of replications. As a practical matter, estimating MSE of the truncated mean is more difficult than the original problem of estimating μ (White, 1997).

Some authors also studied the effect of deletion on confidence interval construction. Kelton (1980) showed for a first-order autoregressive process that replication performed well in terms of coverage when m and d are appropriately chosen. For a fixed value of m, he also observed that deletion ($d > 0$) increased the expected value of the confidence interval half-length. Results in Law (1975, 1977) and Kelton and Law (1985) for the M/M/1 queue indicate that deletion has negligible impact on the coverage of a confidence interval when m is large enough to produce acceptable coverage for μ.

There have been a number of methods proposed for choosing d, ranging from very simple rules-of-thumb to complicated statistical techniques. Some of these rules have already been stated in Section 2.5. Here, we will focus on the techniques that we will use extensively in the remainder of this study.

If the value of d is too large relative to the value of m, then deletion could result in a degradation in coverage. As a rough rule, the length of each replication, beyond the deletion point, would be at least ten times the amount of data deleted. Given this run length, the number of replications should be as many as time permits, up to 25 replications (Kelton, 1986). Whitt (1991) suggests that the appropriate amount to delete should usually be a relatively small portion of the total simulation run length (as low as 5%).

Plotting the cumulative average graph

Different approaches to truncation apply different means to estimate the truncation points and imply different criteria for what makes an observation "representative" of steady-state. Perhaps the most venerable approach is *visualization*. The human faculty for recognizing visual patterns should not be underestimated, and, to the extent practicable in a given study, individual and cumulative observations should be plotted and inspected visually (Kleijnen, 1984).

A deletion point d from a cumulative average graph can be obtained by looking for a point where the curve seems to become nearly horizontal. As can be understood from the criterion statement this is a very subjective way of selecting a truncation point. Since it does not provide any quantitative value to assess the truncation point, deciding on the length of the transient period might differ drastically from one analyst to the other.

Formally stated, given a stochastic output sequence $\{X_i,\ i = 1, 2, \ldots, m\}$ the cumulative average statistic is calculated as

$$\overline{X}_k = \frac{1}{k}\sum_{i=1}^{k} X_i \qquad \text{for } k = 1, 2, \ldots, m \tag{2.20}$$

and $\overline{X}_k$ for $k = 1, 2, \ldots, m$ is plotted against k. A truncation point d is selected visually from the plot where the curve flattens out.

Marginal Standard Error Rules (MSER and MSER-5)

The MSER (White, 1997) and MSER-5 (Spratt, 1998) rules determine the truncation point as the value of d that best balances the tradeoff between improved accuracy (elimination of bias) and decreased precision (reduction in sample size) for the reserved series $X_{d+1}, X_{d+2},\ldots, X_{d+m}$. These methods select a truncation point that minimizes the width of the confidence interval, instead of selecting the truncation point to minimize MSE due to the difficulty in its estimation. They try to mitigate the bias by removing initial observations that are far from the sample mean, but only to the extent this distance is sufficient to compensate for the resulting reduction in sample size in the calculation of the confidence interval half-width.

The determination of the truncation point for a given output sequence $X_1, X_2,\ldots, X_m$. is formally stated as follows:

$$d^* = \underset{m>>d\geq 0}{\arg\min}\left[z_{\alpha/2} \cdot \sqrt{\frac{S^2(d)}{m-d}} \right] \tag{2.21}$$

where $z_{\alpha/2}$ is the value of the unit normal distribution associated with a 100(1-α) percent confidence interval, $S^2(d)$ is the sample variance of the reserved sequence, and m-d is the number of observations in the reserved sequence. Notice that the quantity in square brackets in (2.21) is the half-length of the confidence interval for μ. For a fixed confidence level, $z_{\alpha/2}$ is a constant, and the expression in (2.21) can be rewritten as:

$$d^* = \underset{m>>d\geq 0}{\arg\min}\left[\frac{1}{(m-d)^2}\cdot\sum_{i=d+1}^{m}\left(X_i - \overline{X}(m,d)\right)^2\right] \tag{2.22}$$

For a given output sequence, d^* is determined by solving the unconstrained minimization problem defined in (2.22). While the MSER heuristic applies (2.22) to the raw output series $\{X_i\}$, MSER-*m* instead uses the series of $b = \lfloor m/n \rfloor$ batch averages $\{Z_j\}$, where $\lfloor \cdot \rfloor$is the maximum integer function, and

$$Z_j = \frac{1}{n}\sum_{p=1}^{n} X_{n(j-1)+p} \tag{2.23}$$

Although the authors of the MSER heuristics state that they are trying to minimize the marginal confidence interval half-length, for a fixed confidence level α, the problem reduces to minimizing a very simply statistic, i.e., standard error (*s.e.*) of the estimate.

$$s.e. = \sqrt{\frac{S^2(d)}{m-d}} \tag{2.24}$$

Perhaps the most important advantage of this technique is that it provides quantitative values for truncation point, hence can be an objective criterion. Another advantage is that this statistic is so easy to compute. Even for very large sample sizes it can readily be solved by complete enumeration. However, when investigated critically, the technique involves a crucial problem in estimating the truncation point. In estimating the standard error, it makes use of the sample variance, $S^2(d)$. In fact, it calculates the sample variance from a single simulation output, where we know that the outputs of a single simulation run are sequentially correlated. And this autocorrelation might induce a significant amount of bias in the variance estimation. This means that standard error estimates will also be biased. At first sight, this might provide some skepticism to the analyst regarding the credibility of the heuristic. However, the developers of the heuristic states that the sole purpose in using this statistic is to measure the homogeneity of the truncated series reserved for analysis (White, 2001). That is, the statistic is not used to estimate the precision of any performance measure, rather it is used to see how homogenous are the output sequences. In other words, the MSER heuristics try to observe the behavior of the standard error estimate and detect the truncation point from its behavior. The underlying assumption, which is not explicitly stated by the authors of MSER heuristics, is that the behavior of the standard error estimate of the sequence will approximately remain same regardless of the existence of autocorrelation in the sequence. Another disadvantage of the technique is that it is very sensitive to the existence of outliers (extreme values) in the sequence. Unless extreme values are carefully deleted from a sequence MSER heuristics can behave badly, even backfiring might occur.

3 THE PROPOSED STUDY

The problem of initial transient for nonterminating simulations is discussed in considerable detail and an extensive review of the literature on this problem is given in the previous chapter. The studies conducted in this area, in a very broad sense, can be grouped as the ones that propose a method to mitigate the effects of initialization bias and the ones that compare and give recommendations about the efficiency and effectiveness of the proposed methods. It has been realized that almost none of the studies have analyzed the *behavior* of the transient period with respect to different system parameters.

The purpose of this study is to answer the following research questions:

1. How and in what way is the transient period affected by different system parameters (or factors)?
2. Which factors most significantly affect the transient period?
3. How do the proposed methods to remedy the initialization bias problem comply with each other?

The first question should not be confused with the following one: "How does the initial conditions affect the length of the transient period?" This problem investigates the *effect of different initial conditions* on the length of transient period and is extensively studied in the literature (see, for example, Conway, 1963; Blomqvist, 1970; Madansky, 1976; Kelton, 1980, 1989, 1985; Murray and Kelton 1988a, 1988b; Murray, 1988; Kelton and Law, 1981, 1985; Adlakha and Fishman, 1982; Snell and Schruben, 1985; and others in Table 2.2). However, the problem stated in this study investigates the *effect of different system parameters* on the length of the transient period given a *fixed set of initial* *conditions*. The second question tries to point out the factors, which will be defined in the following sections that have the most significant effect on the length of the transient period. In question three, we try to find out the consistency between two alternative remedial procedures for the initial bias problem, namely, cumulative averages plot and MSER heuristics, by comparing their performances under various experimental conditions.

Specifically, we will try to answer these questions by experimentally analyzing the simulation of manufacturing systems, particularly, serial production lines and job-shop production systems.

3.1 The methodology used in this study

In this section, we outline the environment used for building and generating the output sequences. Furthermore, the implementation of techniques used for the analyses of outputs is discussed. We used AutoMod Ver. 9.1 (see AutoMod User's Manual, 1999), which is one of the most popular software for simulation of manufacturing systems, to build our simulation models.

While comparing the effects of different factors on a specific behavior of interest via real-life experimentation, the experimenter tries to prepare samples as identical to each other as possible except for the factors that are investigated. Successful experimentation is done in this manner, because otherwise it would be impossible to relate the changes in the performance measures of interest to the changes in the factor levels, if there is any difference among the results of experiments. This phenomenon is sometimes called as comparing the like with likes. An excellent text about the design and analysis of experiments is provided by Montgomery (1984).

The above caution about physical experimentation also applies to simulation experiments. In order to make unbiased comparisons between different simulation experiments (or designs), one needs to build the simulation models with a common base and differentiate only for the specified factors.

It is important to remember that simulation experiments use inputs that are themselves random variables. As a result, the outputs (or performance measures) of interest will clearly be random variables, too. Having this fact in mind, one questions the main reason for the changes in the performance measures of a system, if there is any. It might be either due to changes in the factor levels, which is, in fact, the main purpose of experimentation or due to inherent variability of system that results with the violation of the idea of comparing the like with likes.

Every simulation experiment makes use of pseudo-random number generators that produce the necessary data for input to the simulation. The outputs generated by these generators are called pseudorandom numbers, rather than random numbers, because generator itself uses a predetermined algorithm. However, although the numbers are generated by a deterministic algorithm they are viewed as random, because it is unlikely for statistical tests to recognize that they were produced in a deterministic fashion.

In particular, we used the famous Linear Congruential Generator (LCG) in our models, which was first introduced by Lehmer (1951) and is well-discussed in Law and Kelton (2000) and Banks et al (1996). This generator produces an ordered sequence of random numbers $u_1, u_2, \ldots, u_g$, where $0 \le u_i < 1$ and g is the period of the generator and is equal to 2^{31}-1 (see AutoMod User's Manual, 1999). Pseudo-random numbers are controlled through streams that take subsequences from the whole sequence by simply defining a reference point. The subsequences between streams will be independent of each other, since the whole sequence is composed of independent and identically distributed numbers. The use of streams gives a considerable advantage to the analyst. In particular, using the same random number streams for different designs will induce dependence, whereas using different random number streams will induce independence between the simulation replications.

In order to facilitate the idea of comparing like with likes, we used a popular method called common random numbers (CRN), which is also used frequently for the purpose of variance reduction. A detailed discussion of using common random numbers and other variance reduction techniques is given in Law and Kelton (2000). We need complete synchronization to properly implement CRN. For this purpose, we divided the entire random number sequence generated by LCG into several subsequences (streams) and each subsequence that is used for a specific purpose in one design is used exactly for the same purpose in each of the other designs. Tables A.1 and A.2 in Appendix A show the streams with their reference points and their use in serial production and job-shop models, respectively.

For example, "`stream1`" in serial production models is used for generating the operation times on machine 1 for the first replications of the models. The usage of other streams is interpreted in a similar manner.

Table 3.1 Number of random numbers used in each stream for design "312121221".

Usage Purpose	**Run #1**	**Run #2**	**Run #3**	**Run #4**	**Run #5**
M/C #1 Operation Time	38,250	38,358	38,268	38,006	38,368
M/C #2 Operation Time	38,064	38,426	38,166	37,876	38,366
M/C #3 Operation Time	38,442	38,168	38,326	38,444	38,168
Mean Breakdown of M/C #1	20,144	18,786	20,128	19,046	19,876
Mean Breakdown of M/C #2	20,418	19,048	19,380	19,278	20,250
Mean Breakdown of M/C #3	20,120	18,706	20,422	19,516	20,268
Mean Repair Time of M/C #1	19,182	17,668	19,268	18,442	18,840
Mean Repair Time of M/C #2	19,480	18,570	19,134	18,508	19,378
Mean Repair Time of M/C #3	19,218	17,764	19,504	18,684	19,284

The length of each stream is chosen so as to avoid any overlaps between adjacent streams, which otherwise would induce uncontrolled dependence. In choosing the lengths of the streams we made pilot runs to observe how many random numbers are used for each activity in the models given a fixed run length. Table 3.1 lists the random numbers used for a specific activity for a particular pilot run to produce 30,000 parts. (The choice of 30,000 parts as the run length will be discussed in the sequel.) For instance, to generate the operation time of parts on machine #1 38,250 random numbers are used from `stream1` in replication #1. Similarly, 38,358 – 38,268 – 38,006 and 38,368 random numbers are used from `stream11–stream21–stream31` and `stream41` to generate the operation times of parts on machine #1 in replications #2, #3, #4, and #5, respectively (see Table A.1 to view which streams are used for what purpose). The rest of the Table 3.1 is interpreted in a similar manner.

Although we generate 30,000 random variates as the operation time of parts on each machine for each replication, the number of random numbers used for this purpose from each stream is above 38,000 except for the operation times on the second machine for the fourth replication (37,876). This is because of the random variate generation technique used. The distribution of operation times is assumed to be lognormal (see Section 3.3). AutoMod Ver 9.1 uses acceptance/rejection technique to generate random variates from lognormal distribution (AutoMod technical support, 2001). The number of random numbers used by this technique to generate a random variate is itself a random variable. Hence, to generate 30,000 random variates according to a lognormal distribution we need at least 30,000 random numbers, however, it can be any number greater than 30,000. A detailed discussion of random variate generation techniques is given by Law and Kelton (2000). Furthermore, we assume that the mean time between failures and mean repair times are gamma distributed (see Section 3.3). AutoMod Ver 9.1 uses acceptance/rejection technique to generate gamma variates, as well (AutoMod Technical support, 2001). Therefore, the number of random numbers used to generate time between failures and repair times are well above from the number of breakdown and repair events, respectively. Moreover, the number of random numbers used for generating time between failures is always greater than that of repair times (compare the values in the following row pairs in Table 3.1: (5, 8), (6, 9) and (7, 10)). This is due to the fact that a repair event does not occur unless a breakdown event occurs. However, the occurrence of a breakdown event is independent of the breakdowns occurred before.

Since each stream is dedicated to a specific activity and the run length is fixed, the choice of 1,000,000 random numbers for the length of a stream seems reasonable to avoid overlapping between streams and to provide similarity among different designs. By doing so, we become more confident about the effects of system configurations on the performance measures of interest.

We used the method of independent replications to generate the output sequences. Consider Table 3.2 for clarity (notice that it is a slightly modified version of Table 2.1).

Table 3.2 The format of outputs for a particular model.

Simulation	*Observations*							
Replications	O_1	O_2	…	O_d	O_{d+1}	O_{d+2}	…	O_m
R_1	X_{11}	X_{21}	…	X_{d1}	$X_{d+1,1}$	$X_{d+2,1}$	…	X_{m1}
R_2	X_{12}	X_{22}	…	X_{d2}	$X_{d+1,2}$	$X_{d+2,2}$	…	X_{m2}
.	.				.			.
.	.				.			.
.	.				.			.
R_n	X_{1n}	X_{2n}	…	X_{dn}	$X_{d+1,n}$	$X_{d+2,n}$	…	X_{mn}
$\overline{X}_i$	$\overline{X}_1$	$\overline{X}_2$	…	$\overline{X}_d$	$\overline{X}_{d+1}$	$\overline{X}_{d+2}$	…	$\overline{X}_m$

The simulation model is run for a total of *m* observations. Additionally, this is repeated for *n* times. X_{ij} in Table 3.2 is the i^{th} observation in the j^{th} replication (i =1, 2,…, m; j = 1, 2,…,n). The determination of *m* is based on making pilot runs. We wanted to run the simulation long enough to allow all the events in the system to occur considerable number of times so as not to neglect their effect on the system behavior. On the other hand, we did not want to take an unnecessarily long run, which would waste computer time. With its subjective assessment, we decided to take m = 30,000 observations, which, we believe, balances between collection of enough information versus wasting computer time. The index *i* represents the number of the observation in the output sequence where it might also have represented the observations taken at fixed increments of simulated time (Fishman, 1971). The number of replications, *n*, is taken to be 5, which is the recommended number for extensive experimentation (Law and Kelton, 2000).

Table 3.3 Effect of common random numbers (CRN) and independent replications.

Design	Run #	Machine #1		Machine #2		Machine #3	
		MOT*	Utilization	MOT	Utilization	MOT	Utilization
31221	1	1852.46	0.096	16309.86	0.849	1812.13	0.094
	2	1752.80	0.092	16214.55	0.854	1817.69	0.096
	3	1744.34	0.066	23843.18	0.901	1646.72	0.062
	4	1893.12	0.097	16804.45	0.860	1651.07	0.084
	5	1739.49	0.080	18962.42	0.874	1815.79	0.084
	Mean	1796.442	0.086	18426.892	0.868	1748.680	0.084
31222	1	1852.04	0.106	16307.69	0.935	1812.13	0.104
	2	1752.49	0.102	16209.48	0.941	1817.69	0.106
	3	1743.93	0.070	23480.95	0.962	1646.72	0.066
	4	1892.67	0.106	16804.45	0.942	1651.07	0.093
	5	1738.93	0.087	18962.42	0.948	1815.79	0.091
	Mean	1796.442	0.094	18424.998	0.946	1748.680	0.092

* MOT stands for Mean Operation Time and is measured in minutes.

The independence between different replications is provided by devoting different random number streams. Table 3.3 shows the ultimate purpose of using common random numbers and independent replications.

Design 31221 is a 3-staged serial production line containing a 10% bottleneck station in the middle of the line with highly variable processing times (coefficient of variation of the processing times in each machine is 2.5) and zero intermediate buffer capacity (see Section 3.3 and Table C.1 of Appendix C for a more complete definition of model assumptions and design parameters, respectively.) Design 31222 is exactly the same as 31221 except for the buffer capacities. Design 31222 includes intermediate buffers each having a capacity of 10.

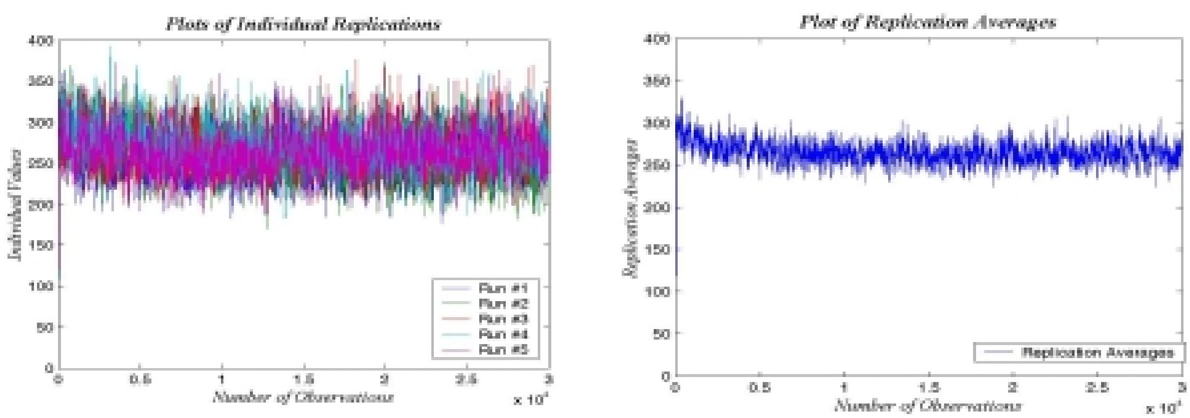
Mean operation times in machines #1, #2, and #3 for each replication are *approximately* the same for designs 31221 and 31222, which is a result of using dedicated common random numbers. In particular, the mean operation time of machine #1 on replication #1 is 1852.46 and 1852.04 minutes for designs 31221 and 31222, respectively (a difference of only 0.02267%). These numbers are obtained by averaging the individual operation times of each part processed in that machine. The maximum deviation between two designs occurs in the mean operation time of machine #1 on replication #5 (a change from 1739.49 minutes to 1738.93 minutes causes 0.03219% deviation). The deviations occur, because of the additional buffer spaces in design 31222. The simulation models terminate as soon as the 30,000th part leaves the last station. Since there are no intermediate storage areas in design 31221, 30,000 operation times are averaged to calculate the mean operation times. However, for design 31222, more than 30,000 parts enter the system, which causes the use of more than 30,000 operation times in calculating the mean operation times for machines #1 and #2. Notice also that the mean operation times on machine #3 for each replication are *exactly* the same for both designs, because exactly 30,000 observations are used to calculate the mean operation time on machine #3 in both designs. Similar results are observed for the breakdown statistics but they are not reported here.

Additionally, the mean operation times of machine #1 in design 31221 is 1852.46 minutes for replication #1, which is totally different from the mean operation times of this machine in the other four replications of the same design. Furthermore, the mean operation times of machine #1 in each of the other four replications differ from each other for design 31221, as well. This is a result of using independent random number streams. Similar results are also observed for machines #2 and #3, and for design 31222, as well. Utilization statistics are provided only to show the difference between the two designs.

Once all the observations for each replication are collected, we average them across replications and obtain $\overline{X}_i$'s for i = 1, 2,…, m, where

$$\overline{X}_i = \frac{1}{n}\sum_{i=1}^{n} X_{ij}, \quad \text{for } i = 1, 2, \ldots, m \tag{3.1}$$

Notice that this average is different from that of (2.14), which was discussed before. Referring to Table 3.2, since the replications produce independent sequences, averaging across replications will give an unbiased estimator of observation O_i, i = 1, 2,…, m. However, the autocorrelation structure of the $\overline{X}_i$'s is still the same as that of the original X_{ij}'s. Therefore, one needs to use equation (2.3) to estimate the variance of the point estimator, which might be obtained by averaging over $\overline{X}_i$'s. In consequence, the sole purpose of using multiple replications in this study is to enhance the ability of determining the end of the warm-up period, since averaging over multiple runs will result in more precise, or less variable, observations. The effect of averaging across replications is shown in Figure 3.1*. The oscillations in the plots of individual replications are apparently higher than that of replication average.



(a) (b)

Figure 3.1 Effect of averaging across several replications on the variability of the process.

Additionally, in Tables D.1 and D.2 of Appendix D the statistics such as hourly throughput, hourly work-in-process, mean flow-time of parts and utilization of each individual machine in the system are given, which are obtained from serial line and job-shop models, respectively.

* Note that outliers in the output sequence are deleted in drawing Figure 3.1.

All of the statistical results discussed above are used in verifying our models, which means that our computer program performs as intended. Additionally, the animation capability of and the existence of a debugger program in AutoMod extended the credibility of our models.

Once our models are built and verified, and all the necessary information are gathered by having appropriate runs we come to the process of analyzing the results for the length of the transient period. Although many statistics have been reported in this study, we used only the flow-time of parts in the system to analyze the length of the transient period phenomena. All of the following analyses have been completed in Matlab Ver 5.3 by writing appropriate computer codes.

We started our analysis with a pilot study by examining the plots of the individual observations against observation number. Although it helped for certain instances, most of the time it was inconclusive due to high variability. Due to extensive experimentation we reported only a few samples of these figures in Appendix F.

In addition to the decrease in the variability of the output sequences attained by independent replications, we used a batching strategy with a batch size of 5 to further smooth the observations. The batching strategy suggests grouping the observations as in the following manner. Suppose we have m observations (O_i) and batch size is given as b, where m is an integer multiple of b, i.e., $m = k \cdot b$ (k is positive integer). Then observations can be grouped as;

$$\underbrace{O_1, O_2, \ldots, O_b}_{\overline{O}_1},\ \underbrace{O_{b+1}, O_{b+2}, \ldots, O_{2b}}_{\overline{O}_2}, \ldots,\ \underbrace{O_{(k-1)b+1}, O_{(k-1)b+2}, \ldots, O_{kb}}_{\overline{O}_k}$$

where,

$$\overline{O}_i = \frac{1}{b} \sum_{j=(i-1)b+1}^{ib} O_j \qquad for\ i = 1,\ 2,\ \ldots,\ k$$

The $\overline{O}_i$'s are called the batched sequence. This kind of batching strategy is extensively used in the literature (see Nelson, 1992, for example) and has nothing to do with the method of batch means, which is a totally different output analysis technique. Since we are simulating the models for a total of 30,000 observations, by using this strategy we will be left with 6,000 (= 30,000/5) observations.

We used two methods for determining the length of the transient period, namely cumulative averages plot and Spratt's (1998) MSER-5 heuristic. In using the cumulative averages plot we apply equation (2.20) with m = 6,000. Once the cumulative average statistics, $\overline{X}_j$, are calculated, we plot the $\overline{X}_j$'s against j for j = 1, 2,..., 6,000.

Conway (1963) cautions about using the cumulative statistics for the purpose of detecting the length of the transient period. He states that such statistics will typically lag behind the current state of the system and their use can cause the discard of unnecessarily great quantities of information. However, our purpose in doing this study is to observe the behavior of the transient period. Though overestimation might occur, cumulative averages will retain the behavior and will not cause any problem in our conclusions.

Welch's moving average is another very popular graphical technique used for detecting the length of the initial transient period. However, a disadvantage of this technique is that the analyst needs to decide on a windows size (w) by trial-and-error. We applied both graphical techniques in the example provided in Section 3.2 and found that Welch's technique does not suggest a significantly different transient period than cumulative averages plot.

Furthermore, our sole purpose is not to determine a single numeric result for the transient period, but to observe the behavior of this period. Even if any of the techniques used in this study includes some error, then this will be reflected to all experiments, which we believe will not disturb the behavior.

Additionally, although cumulative averages plot gives an excellent way of assessing the length of the transient period, a major problem in its use is that it is an informal subjective way of assessment. Even for the same output sequences different analysts might end up with different conclusions about the length of the transient period. Hence, we need a formal statistical procedure in support to the cumulative averages plot, which can detect the length of the transient period. We used MSER-5 to close this gap. In using this approach we apply equation (2.22) to the batched sequences. In fact, MSER-5 simply uses a batching strategy that is discussed above before applying White's (1997) MSER heuristic. A theoretical drawback of this method is its negligence of the autocorrelation structure of the sequence and is discussed in Section 2.6. A practical drawback of this method, which was also discussed in Section 2.6, is its high sensitivity to the existence of outliers in the sequence.

3.2 An Example

We further explain our methodology with an example, in this section. This example is drawn from among hundreds of experiments that have been conducted in this study, which entail all of the discussions in Section 3.1 The details of model assumptions are presented in Section 3.3 This deference does not impose any problem in understanding the following discussions.

In this study, to easily keep track of the parameter levels we gave specific names to each experiment conducted. These names with their corresponding parameter levels are tabulated in Appendix C. The particular system under consideration is a 9-stage serial production line, which is named as "91122" under our naming logic and parameter levels for this design are given in Table C.1 of Appendix C.

Once all replications of the above model are completed the results can be tabulated as in Table 3.4. The values in the table are flow-time statistics in minutes of each part in the system.

The flow-time of each part for individual replications are shown in rows R_1 through R_5. The replication average over 5 replications are shown in $\overline{X}_i$ row. The effect of averaging across 5 replications on the variability of the process was shown in Figure 3.1. All further discussion is based on the replication averages.

Table 3.4 An example output.

Simulation	*Observations*			
Replications	O_1	O_2	…	$O_{30.000}$
R_1	26.88	29.23	...	348.97
R_2	29.37	31.16	...	252.19
R_3	32.09	33.76	...	268.03
R_4	25.86	33.20	...	248.95
R_5	24.51	45.76	...	286.39
$\overline{X}_i$	27.74	34.62	...	280.91

If no detection about the existence of outliers in the sequence is made than the plots of each individual values would look like the one in Figure 3.2. There are 8 outliers in this sequence that conceal the values of other observations, namely observations 8078, 11830, 13689, 15292, 19945, 22198, 24347, and 24369. The values of these observations are so high that all other values seem to coincide the horizontal *x*-axis. For instance, the smallest of these is observation 8078 with a value of 11,478 is approximately 43 times greater than the average (264.8).

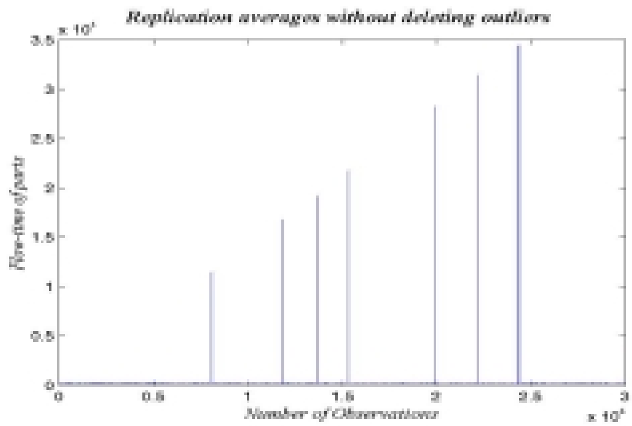


Figure 3.2 A plot of the replication averages without deleting outliers.

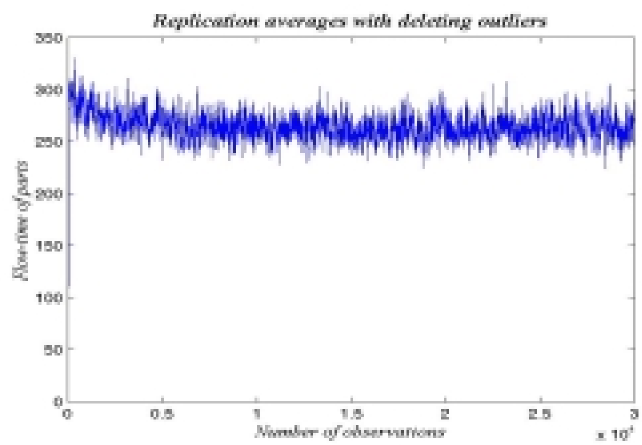


Figure 3.3 A plot of the replication averages with deleting outliers.

Removing these outliers from the sequence leaves only 29,992 observations in the sequence and we obtain the plot of the individual observations as in Figure 3.3. This is a more realistic figure that presents the behavior of the process. Although much effort was given to reduce the variability of the process by averaging across replications, there still exists a considerable amount of variability, which makes it incredibly hard to determine the length of the transient period from this figure. Even the batched sequence would behave in the same way when outliers are deleted.

We continue our analysis by calculating the cumulative averages as discussed in Section 3.1 both for the sequence that contains outliers and for the one that outliers are excluded*. Figure 3.4 illustrates graphically the results of these calculations. We determine the truncation point in a plot, which in turn gives the length of the transient period, by visually investigating the flattening point of the plot. The data up to this flattening point are assumed to represent a transient behavior. As can be seen from the plots, there is no significant change in the length of the transient period when outliers are deleted as compared to the retained case. The only difference between the two sequences is that the one that excludes outliers is shifted slightly towards the x-axis. Both plots suggest approximately the same number of data truncation. This is roughly estimated as 350 observations for the batched sequence. If we were to remember that we applied a batching strategy to the original sequence with a batch size of 5, this will result in the truncation of 2,000 observations from the original sequence. One should keep this fact in mind before continuing with the analyses of the output, but this fact does not cause any harm for our purposes.

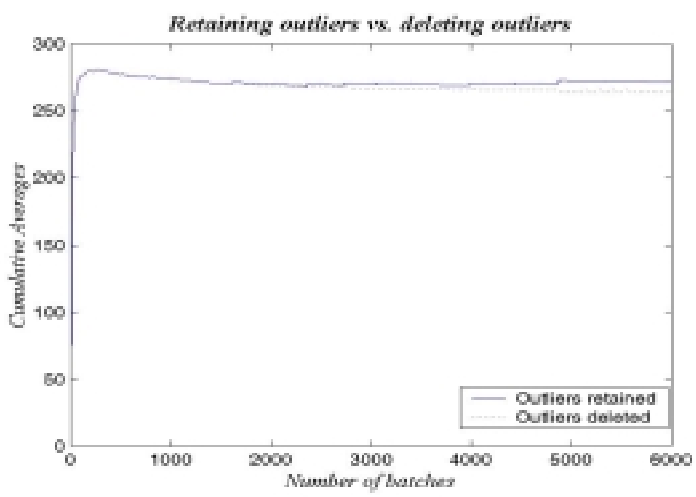
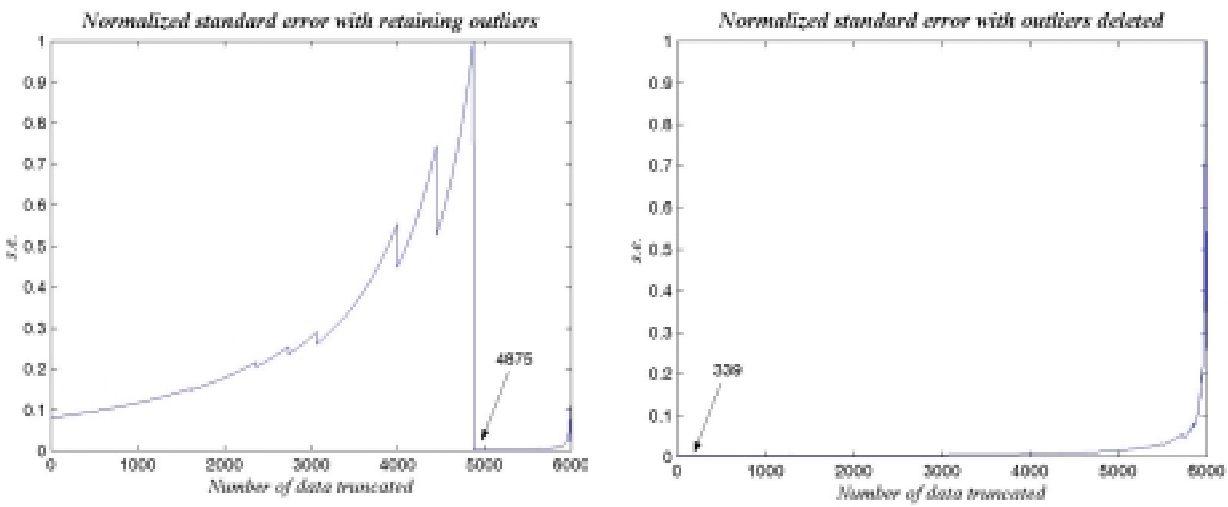


Figure 3.4 Cumulative averages plot for the outliers retained and deleted sequences

* The result of Welch's technique applied to this example for several windows sizes are presented in Figure 4.22. It roughly suggests 300 observations to be in the transient state.

Then, we apply the MSER-5 truncation heuristic to the original sequence both when there are outliers in the sequence and when the outliers are deleted. Figures 3.5(a) and 3.5(b) illustrate graphically the standard error calculations for the no-outliers-deleted and outliers deleted sequences, respectively. Furthermore, Figure 3.6 gives a zoomed version of Figure 3.5(b), which enables to observe the behavior of the standard error statistics more easily. The *x*-axis of these figures shows the number of data deleted in calculating the standard error statistic, while the *y*-axis shows the value of the statistic. The plot starts with an average standard error value when the whole sequence is used, i.e., $d = 0$, and slowly decreases up to some point due to unrepresentative behavior of the data in the initial portion of the sequence. From this point on, the plot changes its direction and the values of the statistics begin increasing. This is due to the decrease in the number of data in the sequence, which behave in a similar fashion (i.e., come from the same probability distribution).



(a) (b)

Figure 3.5 Effect of outliers in MSER calculations.

A more rigorous explanation of the above discussion can be made by rewriting the standard error formula:

$$s.e.(d) = \sqrt{\frac{S^2(n-d)}{n-d}}$$
$$= \sqrt{\frac{1}{(n-d)} \cdot \left[\frac{1}{(n-d-1)} \cdot \sum_{i=1}^{n-d-1} \left(X_i - \overline{X}(n,d) \right)^2 \right]}, \quad 0 \le d \le n-1 \qquad (3.2)$$

where $s.e.(d)$ is the standard error estimate of the whole sequence with d observations deleted from the beginning, $S^2(n\text{-}d)$ is the sample variance of the last n-d observations, and n-d is the number of observations in the retained sequence. This formula can be divided into two parts; the first consisting of $1/(n-d)$ and the second of $(1/n-d-1)\sum_{i=1}^{n-d-1}\left(X_i-\overline{X}(n,d)\right)^2$. The first part is simply the reciprocal of the number of retained data and the second part is the sample variance of the retained sequence. The first portion of the formula has its minimum value when the whole sequence is used, i.e., $d = 0$. Then, it gradually increases as more data are left out, i.e., $0 < d < n$. The behavior of the second part is more complicated. If the whole sequence were to come from the same probability distribution then this part would also have its minimum at $d = 0$, and as d gets larger its value would gradually increase, too, which is due to the decrease in the denominator of the sample variance. If, however, the whole sequence can be separated into two portions where after a certain point d^*, the remaining sequence seems to come from the same stationary distribution then the above result would still be valid. That is, the second part of the formula would gradually increase as d gets larger ($d^* \leq d < n$). On the other hand, for the points that are before d^* in the sequence, the second part of the formula would gradually decrease as d gets larger ($0 \leq d < d^*$). This is because of the fact that the data in the initial portion of the sequence do not come from the same probability distribution as others. As we remove more data from this portion, the distributions of the remaining sequence becomes closer to each other.

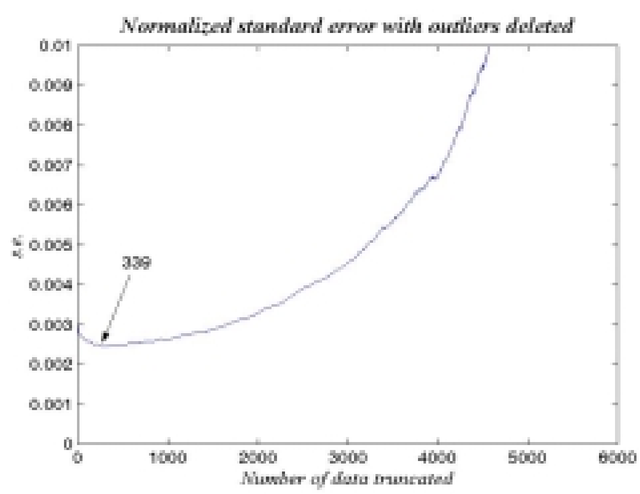


Figure 3.6 The behavior of standard error statistic.

Now, we see from Figures 3.5(a) and 3.5(b) that the standard error statistic gets its minimum value at $d^* = 4875$ and $d^* = 339$, respectively, which suggests the truncation of 4875 and 339 observations from the batched sequence. 4875 seems to be rather too much when the result of the cumulative averages plot is considered. Noting that Figure 3.5(a) is drawn for the sequence that involves outliers, we conclude that 339 is a more reasonable truncation point for this sequence, which also complies with the result of cumulative averages plot.

In determining the length of the transient period in the rest of this study, we follow exactly the same steps described in this example.

3.3 System Considerations and Experimental Design

We consider two types of manufacturing systems. The first system under consideration is a serial production line. A typical serial production line with N stages is shown in Figure 3.7.

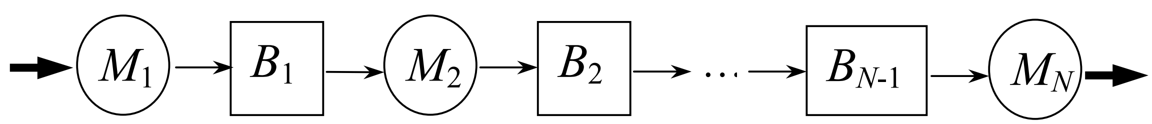


Figure 3.7 A schematic view of N-staged serial production line.

As can be seen from Figure 3.7, the system consists of a sequence of serially arranged machines M_i, $i = 1, 2,\ldots, N,$ with a buffer B_i, $i = 1, 2,\ldots, N-1$, in between two machines. We assume the system works under the following set of assumptions:

1. The system under consideration is an *asynchronous system*. That is, machines within the system have random processing times usually drawn from certain probability distributions, hence they do not have to start and stop their operations at the same instant.
2. Each machine in the system has *mutually independent processing time* distributions.
3. Each *machine* has a maximum *processing capacity* of *one* unit of product at a time and has *internal storage capacity* for that unit.

4. The system is a *saturated system*. That is, there is an infinite supply of raw materials into the system and there is infinite demand for the finished parts.
5. All the *buffers* in the system have *finite storage capacities*, hence the machines, except the first and the last, can starve or get blocked. A machine, M_i, is said to be *blocked* if the completed item in M_i can not be transferred to its downstream buffer, B_i, which occurs when the downstream buffer B_i is full (*blocking-after-service* policy, see Dallery and Gerhswin, 1992). Furthermore, a machine M_i is said to be *starving* if it is ready to process an item but there is either no item to be processed by that machine or the machine is blocked, which occurs when the upstream buffer, B_{i-1}, is empty or the downstream buffer, B_i, is full, respectively.
6. First machine never starves, because there is an infinite supply of raw materials, and the last machine never gets blocked, because there is infinite demand for finished products (also see assumption 4.)
7. The machines are subject to *random failures*, with independent inter-failure time and repair time distributions. The occurrence of a failure event does not depend on how many parts being processed by that machine, rather it is determined by the passage of time (*time-dependent failures* policy, see Buzacott and Hanifin, 1978). If there is a part being processed on a machine at the time of a failure, then the part stays on machine during the repair period and upon completion of the repair its processing is resumed exactly at the point it stopped, i.e., *no rework or scrap*.
8. Machines continue processing unless they are blocked, starved, or in down state.
9. There is only *one type of product* produced by the system, i.e., *no setup times*.
10. A part has to *visit all the machines* in the system *in the given sequence*.
11. The production line assumes *empty and idle initial conditions*. That is, there are no unfinished parts in the buffers and all the machines are idle but ready to operate at the beginning.
12. The system *need not have a steady-state* operating regime.

The second system under consideration is a job-shop production system for which a schematic view is given in Figure 3.8. This system shares many of the assumptions given for the serial line system with slight modifications. Hence, instead of repeating most assumptions we will refer to serial line assumptions when it is valid for the job-shop system, too, and write the ones that are valid for job-shop only. Assumptions 1, 2, 3, 7, 8, 9, 11, and 12 of the serial line system are also valid for the job-shop system, as well. In addition to these, we have the following assumptions:

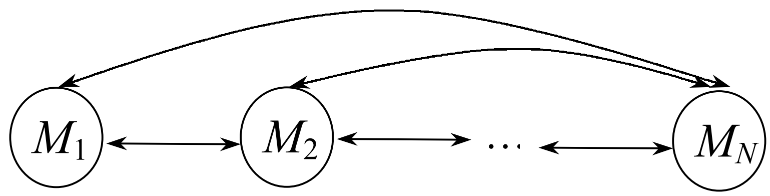


Figure 3.8 A schematic view of *N*-machine job-shop production system.

1. There are *no intermediate storage buffers* in the system.
2. Parts arrive to the system according to a Poisson process. In other words, the interarrival time of parts is exponentially distributed with parameter λ.
3. A part has to visit all the machines in the system, but the processing sequence of a part is not known in advance, rather it is determined randomly. The same machine can not be visited by the same part more than once. Each machine are equally likely to be selected in the sequence.
4. A newly arrived part waits in the system until the first machine in its sequence is available for processing.
5. A machine is blocked if it can not dispose the finished part to the next machine in the processing sequence of the part. Moreover, a part can never block the machine, which is in the last position of its processing sequence.
6. Every machine in the system can starve if there is no part to process.

These systems are extensively studied in the literature (see, for example, Dallery and Gershwin, 1992; Papadopoulos and Heavey, 1996; Altıok, 1997; Buzacott and Shanthikumar, 1992, 1993).

As discussed in section 3.1, the simulation models are developed in AutoMod. Data on five statistics are collected, which are the flow-time of parts, hourly throughput hourly work-in-process, interdeparture time, and utilization of servers. The last four of these performance measures are reported only for the purpose of verifying the simulation models. These summary statistics are reported in Appendix D. We base our analyses on the flow-time of parts. The length of the transient period is investigated under various experimental conditions. Tables 3.5 and 3.6 summarize the factors and their levels in serial line and job-shop systems, respectively.

Table 3.5 Experimental factors and their levels for serial line system.

Factors	**Levels**
System size	3, 9
Load type	Uniform, Bottleneck (10%), bottleneck (20%), bottleneck (99%),
Load level	1, 0.9, 0.5
Processing time coefficient of variation	0.3, 2.5
Processing time variance	0.3, 2.5
Machine type	Reliable, unreliable (90% availability, FBSR[1]), unreliable (90% availability, RBLR[2]), unreliable (80% availability, FBSR), unreliable (80% availability, RBLR), unreliable (50% availability, FBSR), unreliable (50% availability, RBLR)
Buffer capacity	0, 10, 100

1 FBSR: Frequent Breakdown Short Repair Time
2 RBLR: Rare Breakdown Long Repair Time

Table 3.6 Experimental factors and their levels for job-shop system.

Factors	**Levels**
System size	3, 9
Load type	Uniform, bottleneck (5%), bottleneck (10%)
Load level	80%, 50%
Processing time coefficient of variation	0.3, 1.0
Processing time variance	0.3, 1.0
Machine type	Reliable, unreliable (90% availability, FBSR[1]), unreliable (90% availability, RBLR[2]),

1 FBSR: Frequent Breakdown Short Repair Time
2 RBLR: Rare Breakdown Long Repair Time

Two levels are chosen for system size, namely 3 and 9. In serial line system this corresponds to 3- and 9-stage lines, whereas in the job-shop system it corresponds to 3-

and 9-machine systems, respectively. These are, roughly, the most commonly used system size parameters in the literature.

The existence of bottleneck stations (machines) in a system induces considerable problems in terms of the performance measures. The location of bottleneck station(s) is studied by Erel et al (1996). We allow only one station to be bottleneck. And, it is chosen to be the one in the middle of the line for serial line system. A 9-stage line, for example, will have its bottleneck station as the 5^{th} station. For job-shop system the choice of a location for bottleneck machine does not have any meaning, because the layout is not specified. Hence, a bottleneck machine is randomly assigned for job-shop system. Additionally, for both systems, we kept the total workload of the system constant while forming a bottleneck station. In serial line systems, three different levels are chosen for the depth of bottleneck, namely 10%, 20%, and 99%. If we were to generate a 10% bottleneck station for the 3-stage uniform serial line, then we would transfer 10% of the mean processing times of non-bottleneck stations to the bottleneck station. More specifically, if the mean processing times of the machines for the 3-stage serial line in the uniform case were 1-1-1 time units, then its 10% bottleneck counterpart would have mean processing times as 0.9-1.2-0.9 time units for machines 1-2-3, respectively. For the job-shop experiments, only two levels are chosen for the depth of bottleneck, namely, 5% and 10%. We decreased the number of levels from three to two, because the results showed that there occurs a consistent pattern in the outputs. The direction of change in the outputs remains constant, which enables us to make generalized conclusions. Furthermore, The first level is chosen as 5% instead of 10%, because job-shop systems are harder to simulate in terms of computer time (i.e., computer run time increases exponentially with an increase in the depth of bottleneck).

Load level is another factor that is investigated in this study. For the serial line system, its levels represent the mean processing time of machines. The smaller the value of the load level in serial lines the higher loaded the system is. For the job-shop system, load level is adjusted by changing the arrival rate. The higher the value of the load level in job-shop the higher loaded the system (Sabuncuoğlu and Karapınar, 1999). Three and two levels are chosen for the serial line and job-shop systems, respectively.

Knott and Sury (1987) experimentally found that the processing time coefficient of variations ranges between 0.22 and 0.57. We chose 0.3 as the low level of the processing time variability in serial lines and 2.5 as the high level (Erel et al., 1996). For the job-shop system, the low level is also chosen as 0.3, however the high level is chosen as 1.0 because of the longer runtime requirement for these systems (Enns, 2000). In forming bottlenecks, we have to differentiate between variance of processing times and coefficient of variation of processing times, because the values of the mean processing times changes. If the variance is kept constant, then the non-bottleneck stations will have higher coefficient of variation as compared to their uniform counterparts. Similarly, the bottleneck station will have lower coefficient of variation as compared to its uniform counterpart. Table 3.7 illustrates this case for a 3-stage serial line.

Table 3.7 Differentiating between constant PV and constant CV.

Machine	**Uniform System**			**Bottleneck System**					
				Constant PV			**Constant CV**		
	MPT[a]	PV[b]	CV[c]	MPT	PV	CV	MPT	PV	CV
1	1	0.09	0.3	0.9	0.09	0.333	0.9	0.0729	0.3
2	1	0.09	0.3	1.2	0.09	0.25	1.2	0.1296	0.3
3	1	0.09	0.3	0.9	0.09	0.333	0.9	0.0729	0.3

[a] MPT : Mean Processing Time
[b] PV : Processing time Variance
[c] CV : Processing time Coefficient of Variation

The reliability of machines is an important issue that is often neglected in analytical studies. We differentiate between reliable and unreliable systems. Furthermore, we investigate the depth of unreliability by taking three levels for the serial line systems. This is done by setting downtime and uptime parameters so as to achieve a machine efficiency, *e*, of 90%, 80% and 50%. For the reasons discussed above, we chose only one level, i.e., 90% availability, for the job-shop system. Moreover, the type of breakdowns is shown to have significant effect on the performance measures such as throughput and work-in-process inventory (Hopp and Spearman, 2000). They showed that, given the availabilities, a system that experiences frequent breakdowns but short repair times is preferable to a system that experience rare breakdowns but long repair times. Table 3.8 shows the parameters selected for breakdown phenomena (Table B.1 in Appendix B is the detailed version of this).

Table 3.8 Different breakdown scenarios.

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

[a] MTBF: Mean Time Between Failures (in hours)
[b] MRT: Mean Repair Time (in hours)
[c] TST: Total System Time (in hours)

Finally, for serial line systems, the effect of buffer capacities is also investigated. Conway et al. (1988) found that the throughput of serial line systems is not affected significantly if the buffer capacity is increased further beyond six. We have chosen three levels for the buffer capacity, namely, 0, 10, and 100. However, the effect of buffers can not be observed in job-shops, because of the no intermediate buffers assumption.

Additionally, we assume that the processing times on machines for both systems have lognormal distribution. We choose this distribution for the processing times as often used in practice (Law and Kelton, 2000; D’angelo et al., 2000; Kadıpaşaoğlu et al.,2000). Chow (1990) also recommends using a positively skewed distribution for this purpose.

Furthermore, we assume that the machine uptime and downtime has a gamma distribution with shape parameters $\alpha_U = 0.7$ and $\alpha_D = 1.4$, respectively. These parameter values are suggested by Law and Kelton (2000). The scale parameters β_U and β_D for the uptime and downtime distributions are given as:

$$\beta_U = \frac{e\mu_D}{0.7(1-e)} \qquad \beta_D = \frac{\mu_D}{1.4}$$

respectively, where μ_D is the mean downtime specified by the analyst and e is the long-run efficiency of the machines (The derivation of scale parameter formulas are reported in Appendix B). If we let μ_U to be the mean uptime of the machine, then e is defined as:

$$e = \frac{\mu_U}{\mu_D + \mu_U}$$

We use this information in determining the parameters of the uptime and downtime distributions for a specified efficiency level, which forms the levels of the factor “machine type”.

4 RESULTS FOR SERIAL PRODUCTION LINES

We discussed the initial transient problem and remedial approaches to this problem in the preceding sections. We also cleared out the methodology used in this study. The system under consideration in this chapter is a serial arrangement of several machines, which is discussed in considerable detail in Section 3.3. The experimental factors for this system are also discussed in the same section. Hence, it might sometimes necessitate visiting Section 3.3 in order to better follow up the results presented in this section.

There are 7 experimental factors for this system with differing levels each. If full factorial experimentation were to be made than we would need $2 \times 4 \times 3 \times 2 \times 2 \times 7 \times 3 = 2{,}016$ different design points. Additionally, we noted that we make 5 replications for each design point, which in turn would require $2{,}016 \times 5 = 10{,}080$ different simulation runs. However, after recognition of a pattern in the outputs, we decided not to experiment with all design points, which otherwise would be a waste of time and other resources. In summary, we have done 363 experiments (18%), for which we present the results below.

The structural framework in the presentation of outputs

Before proceeding with the results, we outline the structural relationships among the figures presented in this section. The figures can be viewed as a 4 × 2 matrix format. That is, there are 4 rows and 2 columns in *most* of the figures, with each row-column intersection containing a small figure. To ease the job of following the relationships between different designs, we structure the figures in the following manner.

The first column of each figure is composed of the designs with low variable (either CV or PV) processing times, whereas the second column is composed of the designs with highly variable (either CV or PV) processing times. Moreover, the rows within a figure present the designs with different workload distributions. The first row is composed of the designs where the total workload is distributed uniformly among machines. The second row is composed of designs that include a 10% bottleneck station in the middle of the line, i.e., the total work processing time is distributed to the machines in such a manner that the machine in the middle of the line takes 10% of the processing times of the other machines. Similarly, the third and fourth rows are composed of designs that involve a 20% and 99% bottleneck station, respectively. Furthermore, each of the small figures in a row-column intersection includes three separate lines. These lines correspond to the cumulative averages plot of flow-time statistic for the designs that differ only in buffer capacities. The name of the design is shown next to the corresponding plots. (Note that the complete list of design names with their parameter levels is presented in Appendix C.) Additionally, the numbers in parentheses next to the design names are the truncation points suggested by the MSER-5 heuristic. The *x*-axis of each figure shows the number of observations in the sequence, whereas the *y*-axis shows the flow-time of parts in minutes.

The following example clarifies the discussion. The three lines in Figure 4.1 (a) correspond to designs `31111`, `31112`, and `31114`. The truncation points suggested by MSER-5 heuristic for these designs are 8, 33, and 5999, respectively. The designs in Figures 4.1 (a) (i.e., `31111`, `31112`, and `31114`) and 4.1 (b) (i.e., `31211`, `31212`, and `31214`) are exactly the same designs having a buffer capacity of 0, 10, and 100, respectively, except the variability. The former designs have a CV of 0.3 whereas the latter ones have a CV of 2.5. The figures in other rows are interpreted similarly.

Additionally, the designs in Figures 4.1 (a) and 4.1 (c) differ only on the distribution of total processing time to the machines. Processing times of the machines in Figure 4.1 (a) are 1-1-1 minutes, whereas it is 0.9-1.2-0.9 minutes for the machines in Figure 4.1 (c) (10% bottleneck machine). The depth of the bottleneck is further increased to 20% and 99% in Figures 4.1 (e) and 4.1 (g) resulting in a workload distribution of 0.8-1.4-0.8 and 0.01-2.98-0.01, respectively. The same pattern is followed in the second column, as well.

Most of the figures in this section share the above discussed structure. However, since we did not perform a full factorial experimentation due to a recognized pattern, some of the figures are left incomplete in the sense that they do not have 4 × 2 matrix structure. Nevertheless, the basic structure of these figures also complies with the above discussion.

4.1 The effect of buffer capacity

The experimental results show that *buffer capacity has significant negative effect on the length of the transient period*. This counterintuitive result is observed by viewing the individual plots in each of the small figures in Figures 4.1 through 4.19*. Or, with the language of the matrix structure discussed above, this result is apparent by examining the plots in each cell of the matrix for each figure.

Although increasing buffer capacities positively affects many of the performance measures in serial lines, such as throughput of the system, interdeparture time variability, utilization of the machines (see Appendix D), its effect on the length of the transient period is negative. In other words, the length of the transient period increases, as there is an increase in the capacity of the buffers. This is mainly due to the existence of more space availability in a system with more buffer capacity. Such a system needs more time to fill all the spaces, which is an indicator of steady-state.

* The scale of these figures might differ from each other. We present the figures with same scales in Appendix E. However, due to extremely large values for the *y*-axis, those figures in Appendix E do not all have the same scale, as well.

For instance, consider the 3-stage serial line containing all reliable machines with a total processing time of 3 minutes per job, which is uniformly distributed among all machines and the CV of the processing times is 2.5. The results for these designs are presented in Figure 4.1 (b). The cumulative averages plots suggest the end of the transient period as the 1500, 1200, and 1000th observation for designs `31211`, `31212`, and `31214`, respectively. The buffer capacities in these designs increase from 0 in design `31211` to 10 in design `31212`, and further to 100 in design `31214`. The truncation points suggested by the MSER-5 heuristic for these designs are 1167, 1169, and 1187, respectively, which also comply with the results of the cumulative averages plots.

We make exactly the same observation for other system configurations (i.e., systems containing bottleneck machine, systems with low variable processing times, longer lines, and unreliable systems).

4.2 The effect of processing time variability

We have distinguished between two alternative measures of the variability, namely, the variance (PV) and the coefficient of variation (CV). The experimental results for the effect of variability measured by CV on the 3-stage reliable serial line are shown in Figures 4.1, 4.2, and 4.3, and results for the unreliable case are shown in Figures 4.7, 4.8, and 4.9. The effect of variability measured by PV for the same system considerations are shown in Figures 4.4, 4.5, and 4.6 for the reliable case, and in Figures 4.10, 4.11, and 4.12 for the unreliable case.

As expected *variability of the processing times of the machines has significant negative effect on the length of the transient period* (i.e., transient period increases as variability increases.) Variability of the processing times of the machines contributes to the overall system variability, which indeed is the determination criterion for achieving steady-state. In other words, it is assumed that transient period ends when the performance measure reaches approximately a constant value (minimum variability). Consider Figures 4.1 (c) and 4.1(d) for low variable and highly variable systems, respectively. According to cumulative averages plot, the system with high buffer capacities in the low variable case, i.e., design `31124`, reaches steady-state at the 500th

observation whereas the corresponding system in the highly variable case, i.e., design `31224`, reaches steady-state at the 4000th observation. MSER-5 heuristic also complies with the cumulative averages findings by suggesting a truncation amount of 271 and 5970 observations for designs `31124` and `31224`, respectively. Similar observations can be made for the rest of Figure 4.1.

Although, scaling factor is in the favor of graphs that are in the first column of the matrix, that is, they are more detailed, it is still apparent that the transient period takes longer time for the systems on the second column. This observation is valid both for the systems having no intermediate buffers and the ones having intermediate storage areas. Additionally, this conclusion also holds for highly loaded systems (Figures 4.2 and 4.3), for the variability measured by PV (Figures 4.4 through 4.6), for unreliable systems (Figures 4.7 through 4.17), and for longer lines (Figures 4.18 and 4.19), too.

There is one exception to this discussion. Design `31114` in Figure 4.1 (a) and its counterparts in Figures 4.2 (a), 4.3 (a), …, 4.6 (a) seems to have a longer transient period than their highly variable versions, i.e., design `31214` in Figure 4.1 (b) and its counterparts in Figures 4.2 (b), 4.3 (b), …, 4.6 (b). This is mainly due to the special structure of these designs. It is important to recognize that design `31114` and its counterparts are unstable systems (in the flow-time sense) due to the *large buffer spaces* but short and uniform processing times. This causes the flow-time statistic to increase consistently. This can be explained in more detail as follows.

The first column of Figure 4.20 plots the "number of jobs in system" for 3-stage reliable serial line with low variable processing times that is distributed uniformly among machines. Their highly variable versions are presented in the second column of Figure 4.20. The designs in the first row of Figure 4.20 (i.e., `31114` and `31214`) have a buffer capacity of 100. Those in the second (i.e., `31112` and `31212`) and third (i.e., `31111` and `31211`) rows have a buffer capacity of 10 and 0, respectively. The *x*-axes in Figure 4.20 show the number of jobs completed by the system, and *y*-axes show the number of jobs currently either being processed or waiting in buffers. Since there are no available intermediate storage areas in designs `31111` and `31211` the number in system statistic for both low and highly variable designs becomes stable from the beginning of the simulation, i.e., there is no accumulation of jobs in the system (see Figures 4.20 (e) and

4.20 (f)). Almost the same kind of behavior is also observed for the designs that contain intermediate storage areas of 10 units (see Figures 4.20 (c) and 4.20 (d)). The only difference in these figures is the degree of variability. This means that although there are intermediate storage areas, this is not large enough to allow great accumulation of jobs. Figure 4.20 (b) that present the results for highly variable system containing a buffer capacity of 100 units also show the same kind of behavior with the only exception of high amount of variability. However, its low variable counter part, i.e., design 31114, which is presented in Figure 4.20 (a) consistently increases up to 10,000th observation and then becomes stable. The cumulative averages plot of number in system statistic for designs 31114 and 31214 are presented in Figure 4.21 (a) and 4.21 (b), respectively. Apparently, 31214 stabilize earlier than 31114 in terms of number in system. Hence, the consistent increase in number of jobs residing in system causes the flow-time statistics for design 31114 to be never stable for the given run length. This can be generalized to the similarly behaving designs in Figures 4.2 (a) through 4.6 (a).

In short, variability is found to be the most significant factor affecting the transient period. The higher the variability of the machines measured either by CV or by PV, the higher the overall system variability is, the more coupling events between the machines, hence longer the transient period.

4.3 The effect of line length

The results indicate that *line length also has a significant negative effect on the length of the transient period*. As can be seen from the comparison of systems with all reliable machines, increasing line length in the system slightly increases the length of the transient period (each individual plot in Figure 4.1 (a) is compared to that of Figure 4.18(a), 4.1 (b) to 4.18 (b), and so on.) In case of considering the systems with a buffer capacity of 100 units, it is seen that the transient period increases as the line length increases. For example, design 31124 in Figure 4.1 (c) reaches steady-state at the 271st observation whereas its counterpart, i.e., design 91124, in Figure 4.18 (c) reaches steady-state at the 290th observation. The systems with medium and no buffer capacities also confirm to the fact that as the number of stations in the line increases, the length of the transient period also increases.

The inverse effect of line length is mainly due to more coupling in larger systems. The higher the amount of coupling in a system, the harder the system reaches steady-state. This result can also be explained by the following analogy. If the process of achieving steady-state can be viewed as warming-up (or heating) a room (or a building) by several stoves then the determination of the length of the transient period can be viewed as determining the time required to warm-up the stoves to heat the entire room or the building. The short lines can be viewed as small buildings for which the heating of the entire building can be achieved by warming only a few stoves. However, for larger buildings to be heated entirely many more stoves are needed and their warming period will require much energy and time. This results with the conclusion that in order a system to reach steady-state, all the entities of the system would reach steady-state collectively. Since there are more entities in a longer line than shorter ones it will take longer time to reach steady-state for larger system when compared to shorter ones. Similar results can be seen for the systems with all unreliable machines (see Figures 4.13 with their counterparts in Figure 4.19.)

4.4 The effect of distribution of system load

By the distribution of system load we mean the allocation of total processing times among the machines. If any of the machines receive more processing time than the others, then that machine automatically becomes the bottleneck station and controls (or dominates) the flows of jobs in the system. We investigate the effect of bottlenecks on the length of the transient period in two cases, namely the constant CV case and the constant PV case. Furthermore, in each case, we present the results for low and high variability separately. Before presenting the results we further note the following observation.

Observation 1: "*As we transfer processing times from other machines to a single machine, we are in essence moving towards a system that is smaller in size. Hence, also considering the results of line length discussed above, it is expected for the length of the* *transient period to decrease as the depth of the bottleneck is increased given a constant workload.*"

For example, consider a 3-stage uniform system with mean processing times given as 1-1-1 minutes. If we form a 99% bottleneck station by transferring the processing times of the outer stations to the middle one we will obtain a 3-stage system with mean processing times given as 0.01-2.98-0.01. The mean processing times of the 1st and 3rd stations are so small when compared to that of the 2nd station that they can even be neglected. Hence, the 99% bottleneck station can be viewed as a shorter line than its uniform counterpart.

We first consider the constant CV case. It can further be divided into two subclasses, namely the low CV and the high CV. Figures 4.1 (a), (c), (e), and (g) show the results for 3-stage reliable system having a CV of 0.3 with a total processing time of 3 minutes per job, which is distributed uniformly, unevenly with 10% bottleneck, 20% bottleneck, and 99% bottleneck, respectively. As we move from Figure 4.1 (a) to 4.1 (g) the cumulative averages plots suggest *a slight decrease in the length of the transient period*. This is also valid for the MSER-5 statistics. Hence, the results are consistent with *Observation* 1, which states that as the domination of the bottleneck station increases the size of the system decreases and eventually the transient period gets shorter. In this situation, the most important entity in the system becomes the bottleneck station and its arrival to steady-state results in the entire system's arrival to steady-state. Remembering the stove analogy, heating the biggest stove in the building is more important than heating smaller ones to heat the entire building. Similar results are observed for the highly loaded systems (Figures 4.2 (a), (c), (e), and (g), and Figures 4.3 (a), (c), (e), and (g)), and for unreliable systems (Figures 4.7 through 4.12.)

The experimental results for the *high CV case* of the above example are presented in Figures 4.1 (b), (d), (f), and (h). As we move from Figure 4.1 (b) to 4.1 (h) the cumulative averages plots suggest *an increase in the length of the transient period*, which is also confirmed by the MSER-5 statistics. This finding is just the opposite of the result obtained from the low CV case. Notice that in order to keep the CV constant we need to increase the variance (or PV) of bottleneck station since its processing time is higher. Recall that in Section 4.2 we have found that the increase in the variability would significantly increase the length of the transient period. In this section, *Observation* 1 suggests a decrease in the transient period. In the final analysis, the negative effect of variability dominates the positive effect of system size. Therefore, *in the high CV case increasing the depth of the bottleneck also increases the length of the transient period.* It may also be argued that this should also hold for the low CV case, but it should be noticed that the CV in its low level is very low. The results for the high CV case also hold for highly loaded systems (Figures 4.2 (b), (d), (f), and (h), and Figures 4.3 (b), (d), (f), and (h)), and for unreliable systems (Figures 4.7 through 4.12.), as well.

Next, we consider the constant PV case. This can also be investigated in two subclasses as low PV and high PV. Figures 4.4 (a), (c), (e), and (g) show the results for the above discussed example with the only difference that PV is set to 0.3. The decrease in the system size when moved from uniform to bottleneck systems is valid for this case, too (recall *Observation* 1). The cumulative averages plots and the MSER-5 heuristic show *a slight decrease in the length of the transient period*.

However, in the *high PV case*, which are shown in Figures 4.4 (b), (d), (f) and (h), neither the cumulative averages plots nor the MSER-5 statistics suggest a change in the length of the transient period. *Keeping the PV constant in its high level causes no change in the length of the transient period*. An expected behavior due to *Observation* 1 is not observed from the results. The reason is that although the CV is decreased by keeping the PV constant, its value (i.e., 2.5) is still high enough to compensate for any change due to a change in the system size. These results are also valid for highly loaded systems (Figures 4.5 and 4.6) and for unreliable systems (Figures 4.10 through 4.12).

4.5 The effect of system load level

In this section, we first identify what we mean by system load level. Then, we make two observations that help explain the results. Afterwards, we continue with presenting the results. In presenting the results, we distinguish between constant CV and constant PV cases. We investigate the results for low and high variability separately for each case.

System load level is the factor that determines the total work content (TWK) of a system. The higher the TWK of a system the higher loaded a system is, because increasing the TWK causes the system to work faster than the one that has lower TWK. In other words, increasing the TWK will cause the system to process more parts per unit time. From programming point of view, we increase the system load by decreasing the total processing time per job. For instance, consider a 3-stage serial line with a total processing time of 3 minutes per job, which is distributed uniformly among machines, i.e., 1-1-1 minutes. Also, consider a second system with a total processing time of 1.5 minutes per job, which is distributed uniformly, too, i.e., 0.5-0.5-0.5 minutes for each machine. The second system will clearly process more parts per unit time than the first system. Then, it can be stated that a system with smaller total processing times is a zipped version of the system with greater total processing times. This result yields the following observation about the length of the transient period.

Observation 2: "*Since a highly loaded system will process more parts per unit time, the buffers in this system will fill up faster, which in turn will cause a shorter transient period.*"

On the other hand, a second observation about the effect of the change on the load level is as follows.

Observation 3: "*Increasing the load level of a system in essence causes an increase in the congestion level of the system. The increase in congestion level results with more interaction among system entities, which in turn causes more coupling events and an increase in the length of transient period.*"

The analyses of the results for system load level are investigated in two cases as constant CV and constant PV. We first consider the constant CV case. Figures 4.1, 4.2, and 4.3 show the experimental results for a 3-staged reliable serial line with a total processing time of 3, 2.7, and 1.5 minutes per job, respectively, for the constant CV case. For the *high CV case* (i.e., the graphs in the second column of each figure matrix) cumulative averages plots suggest *a decrease in the length of the transient period* as we move from the system with a total work content of 3 minutes to the one with 2.7 minutes and further to the one with 1.5 minutes. This decrease is most apparent when 3- and 1.5-minutes systems are compared. This observation is also supported by the MSER-5 statistics. For instance, consider design 31214 in Figure 4.1 (b), which has a total processing time of 3 minutes per job. The cumulative averages plot and the MSER-5 heuristic suggest the truncation point as 1600 and 1187, respectively. The truncation point for its counterpart, (i.e., design 31244 in Figure 4.2 (b) that has a total processing time of 2.7 minutes per job) is found as 1500 and 1187 by the cumulative averages plot and the MSER-5 heuristic, respectively. If we further reduce total processing time per job down to 1.5 minutes, the truncation point for that design, i.e., design 31274 in Figure 4.3(b), is found as 1000 and 374 by the cumulative averages plot and the MSER-5 heuristic. We see the effect of *Observation* 2, which states that increasing the load level of a system causes the buffers to fill faster and the transient period becomes shorter. *Observation* 3 might also have shown its effect, but as noted earlier variability is the most significant factor that affects the length of the transient period. Hence, the increase in transient period caused by the increase in the congestion level of the system is dominated by the decrease in transient period caused by the decrease in the variability of the system. Similar observations are made for other designs containing high CV processing times. The results also hold for unreliable systems (Figures 4.7, 4.8, and 4.9), as well.

Now, we continue with *low CV case*. The first column of Figures 4.1, 4.2, and 4.3 show the results for the low CV case. The effect of *Observation* 3 is more apparent in this case, because the variability of the system is too low. Hence, the decrease in transient period caused by the decrease in the variability of the system is dominated by the increase in transient period caused by the increase in the congestion level of the system. In other words, *increasing the load level of a system in the low CV case increases the length of the transient period*, however only slightly. This observation is not apparent from the cumulative averages plots, but from the MSER-5 statistics. Consider, for instance, design 31124 in Figure 4.1 (c) that has a total processing time of 3 minutes per job, for which MSER-5 suggests 271 observations to be truncated. Designs 31154 and 31184 in Figures 4.2 (c) and 4.3 (c) that have a total processing time of 2.7 and 1.5 minutes per job, have 273 and 277 observations to be truncated. Other low CV designs also confirm these results. The results are also valid for unreliable systems (Figures 4.7, 4.8, and 4.9), as well.

Next, we consider the *high PV case* whose results are presented in the second column of Figures 4.4, 4.5, and 4.6 for the 3-, 2.7-, and 1.5-minutes systems, respectively. Both the cumulative averages plots and the MSER-5 statistics suggest *no change in the length of the transient period as the load level is increased*. Neither *Observation* 2 nor *Observation* 3 shows their effect, because of the dominance of high variability. The difference of constant PV from the constant CV case is that the variance of the processing times are kept constant in constant PV, whereas there occurs a change in the variances in the constant CV case. Keeping the variance constant at its high level, i.e., 2.5, causes no change in the length of the transient period. Cumulative averages plots and MSER-5 heuristics suggest truncating 1600-1600-1500 and 1187-1187-1187 observations for designs `31714-31744-31774`, respectively. This result holds for other reliable high PV designs and unreliable designs (Figures 4.10, 4.11, and 4.12), as well.

The results for the *low PV*-reliable case are shown in the first column of Figures 4.4, 4.5, and 4.6. There occurs a *slight increase on the length of the transient period*, as we move from 3-minute reliable system to 2.7-minute and further to 1.5-minute systems. The explanation of this result is exactly the same as the low CV case. That is, *Observation* 3 becomes dominant, and the increase in the congestion level of the system causes a slight increase in the length of the transient period, which is only apparent from the MSER-5 statistics. For instance, designs `31624`, `31654`, and `31684` in Figures 4.4 (c), 4.5 (c), and 4.6 (c) reach steady-state at the 231, 233, and 237th item, respectively. Unreliable designs also confirm the above finding (Figures 4.10, 4.11, and 4.12).

4.6 The effect of reliability

The reliability issue has been investigated from two aspects; first is the level of availability and the second is the type of breakdowns. Availability is the long-run ratio of uptime of the machines to the total system time. And, for the type of breakdowns we consider two extreme case: frequent breakdowns/short repair times and rare breakdowns/long repair times.

Firstly, we consider the existence of unreliable machines. Figures 4.7 through 4.12 present the results for 3-stage serial line containing machines that are available only for 90% of the total time, whose reliable counterparts are also given in Figures 4.1 through 4.6, respectively. (Figures 4.7 (h) and 4.8(h) are missing, because we did not allow a computer runtime of more than 1 day, where approximately 2500 parts –less then 10%– were simulated for these designs during this time.) For the *highly variable case* (either measured by CV or PV), *the length of the transient period remains same* for each design, which can be observed by comparing the second columns of each figure, e.g., the designs in the second column of Figure 4.7 should be compared to that of Figure 4.1. For instance, design `31214` in Figure 4.1 (b), reaches steady-state at the 1187th observation, whereas its unreliable version, i.e., design `312141221` in Figure 4.7 (b) reaches exactly at the same observation, according to MSER-5 heuristic. Cumulative averages plot also confirm this finding. This result can be explained with the following analogy: If the variability of a sequence can be viewed as a series of waves in a sea, then a highly variable sequence can be viewed as a sequence of highly wavy ocean. Waves that are generated by an artificial source will have no effect in the big ocean unless the source is very powerful. By allowing the machines to fail we are in effect introducing additional variability to system. However, the variability introduced by breakdowns is so small when compared to the original variability of the system that its effect is negligible. Hence, we can not observe any change in the length of the transient period for the highly variable case. Same result is observed for other highly variable 3- and 9-stage designs, too.

The effect of reliability in the *low variable case* is somewhat more complicated. If we compare first columns of Figures 4.1 through 4.6 (reliable designs) to Figures 4.7 through 4.12 (unreliable designs), we observe *a slight decrease in the length of the transient period* (the type of unreliability in these designs is chosen as *frequent breakdowns/short repair times*). However, this decrease is not apparent from the cumulative averages plot, but from the MSER-5 heuristic. For instance, design `31124` in Figure 4.1 (c) ends its transient period at the 271st observation, whereas its unreliable counterpart, i.e., design `311241221` in Figure 4.7 (c), ends its transient period at the 29th observation. If, on the other hand, we were to compare the unreliable design that works as *rare breakdown/long repair time*, we would observe *a slight increase in the length of the transient period* (compare the first column of Figure 4.1 to that of Figure 4.13). Allowing breakdown events has twofold effect on a system. The first, which is discussed in the previous paragraph, is introduction of additional variability to the system. On the other hand, breakdown events slow down a system, which in turn causes the buffers to fill up earlier. Referring to the effect of buffers, a system that fills its buffers faster will reach steady-state earlier. Therefore, we conclude that for the frequent breakdown/short repair time case the negative effect of the variability introduced by breakdown events is negligible when compared to the positive effect of the buffers. This result is observed for other 3-stage designs and 9-stage designs (compare Figure 4.18 and 4.19), as well. However, for the rare breakdown/long repair time case, the effect if variability introduced by breakdown events dominates the effect of buffers.

Next, we consider increasing the *depth of breakdowns* from 90% availability to 80% and further to 50%. Figures 4.7, 4.14, and 4.15 present the results for 90%, 80%, and 50% availability cases, respectively. For the *high variability case*, the CV of processing times is the dominant factor as in previous discussions. Hence, there is *no change in the length of the transient period*. The MSER-5 heuristic suggests truncating 1187 observations from designs `312141221`, `312141223`, and `312141227`, which differ only for the availability of machines and can be seen in Figures 4.7 (b), 4.14 (b), and 4.15 (b), respectively. However, in the *low variability case*, there is *a slight increase in the length of the transient period*, which is due to the increase in the variability introduced by breakdown events. That is, the more breakdown events, the higher variability introduced. This can be seen in Figures 4.7 (a), 4.14 (a) and 4.15 (a). Other 3-stage designs also confirm this result.

Finally, we consider the *type of breakdown* events. Figures 4.7 and 4.13 illustrate the results for frequent but short and rare but long breakdowns, respectively, when the machine availability is 90%. Similarly, Figures 4.14 and 4.16 show the results for 80% availability, and Figures 4.15 and 4.17 for 50% availability. Experimental results show that in the *high variability case* the situation is similar to the previous ones, i.e., there is *no change in the length of the transient period* between two breakdown types, which is demonstrated both by the cumulative averages plots and the MSER-5 statistics. However, for the *low variability case*, the results show that *rare but long breakdowns attain a longer transient period than frequent but short breakdowns*. This due to the increase in the variability inserted by breakdown events. For instance, designs 311141221 and 311141222 that are based on 90% availability are found to have a transient period of 312 and 374 observations according to the MSER-5 heuristic (see Figures 4.7 (a) and 4.13 (a)). The results are similar for the designs that are 80% and 50% available.

4.7 The effect of utilization

In this study, we also investigated if there is any relation between the length of the transient period and the utilization level of a system. However, our results indicated that *there is no direct relation* between these two measures. The following example illustrates this case.

Table 4.1 Effect of utilization on the length of the transient period.

Design	Average ρ^1	Length of T_p^2	Change in ρ	Change in T_p
31121	0.753	5	-	-
31124	0.814	271	Increase	Increase
31181	0.864	5	Increase	No change
31221	0.346	2302	Decrease	Increase
311a1	0.356	2	Decrease	Decrease

[1] ρ : Utilization
[2] T_p : Transient period

By using the previous results and the ones reported in Appendix D we could find four counter examples. The results are tabulated in Table 4.1 for which the length of transient period is determined by the MSER-5 heuristic. First consider a design, say 31121 (3-stage reliable serial line containing 10% bottleneck station and having no intermediate buffers), for which the transient period and average system utilization statistics are determined as 5 and 75.3%, respectively.

By adding additional buffer capacities to this design we obtain design 31124 (buffer capacity is increase from 0 to 100), which is 8% more utilized than 31121 and has a transient period of length 271 observations. The increase in the utilization of the system causes an increase in the length of the transient period, as well.

If we increase the load level of the system by decreasing the total processing time per job from 3 to 1.5 minutes, we obtain design 31181, which is 15% more utilized than

31121. However, this increase in utilization caused no change in the length of the transient period (i.e., the transient period for design 31181 ends at the 5th observation).

Then, we increase the variability of the processing times and obtain design 31221, which is 54% less utilized then 31121. Although this design is significantly less utilized than 31121, we observe a significant increase in the length of the transient period (from 5 to 2302).

Finally, by forming a 99% bottleneck station we obtain design 311a1, which is 53% less utilized than 31121. However, this decrease caused a decrease in the length of transient period.

In conclusion, we don't have a direct relation between the length of the transient period and the utilization level of the system.

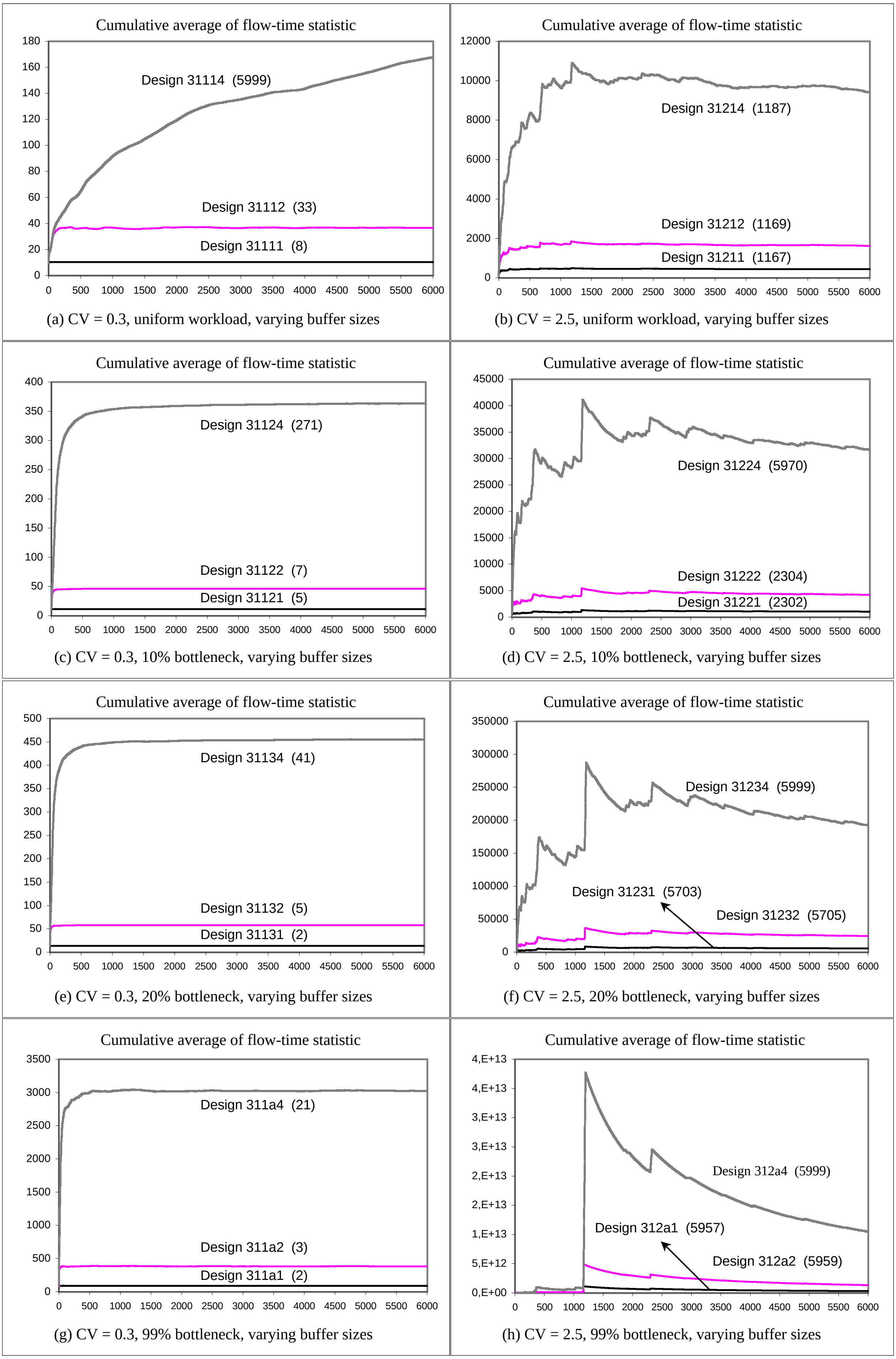


Figure 4.1 Experimental results for 3-stage serial line containing all reliable machines with a total processing time of 3 minutes per job.

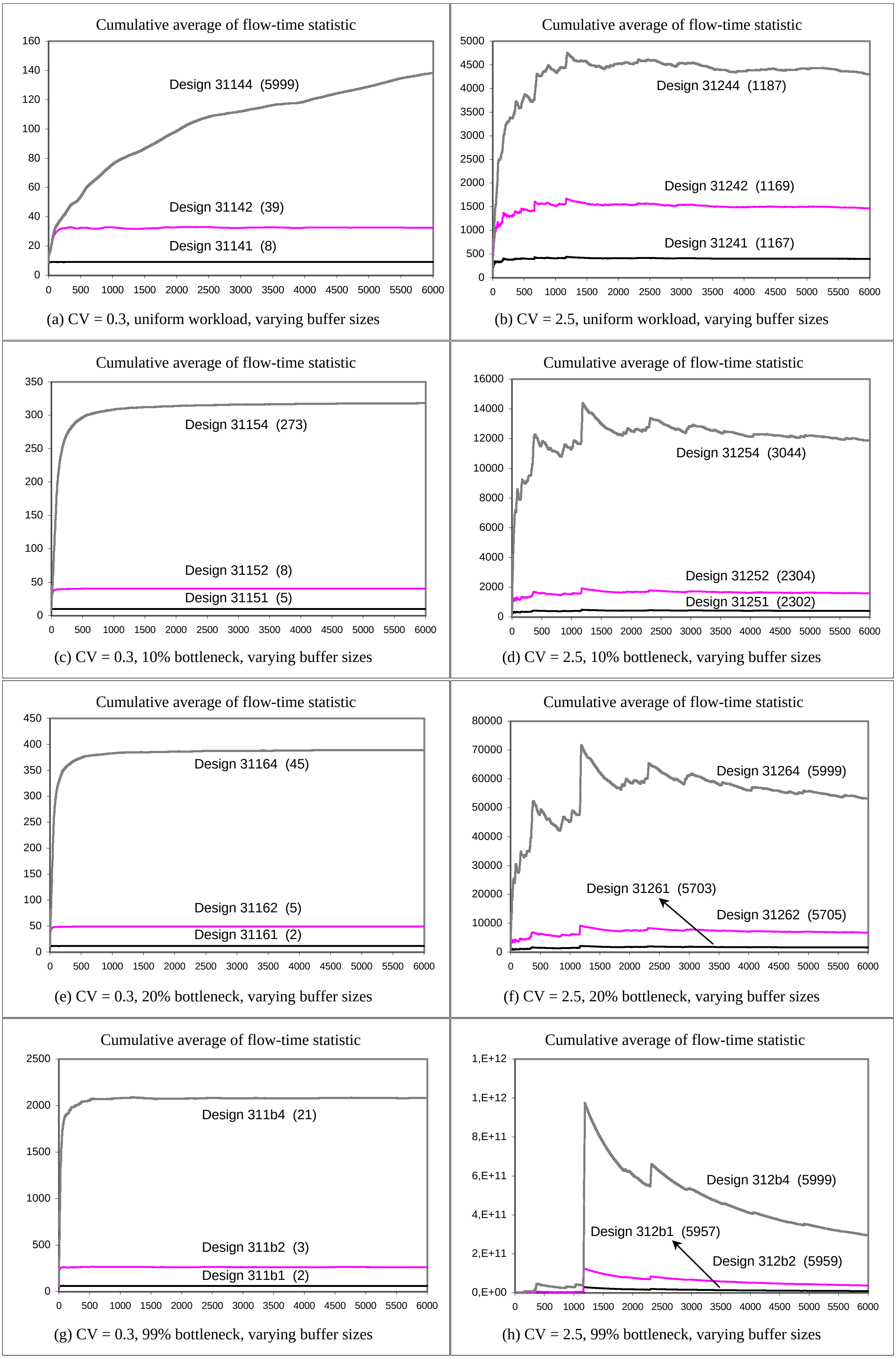
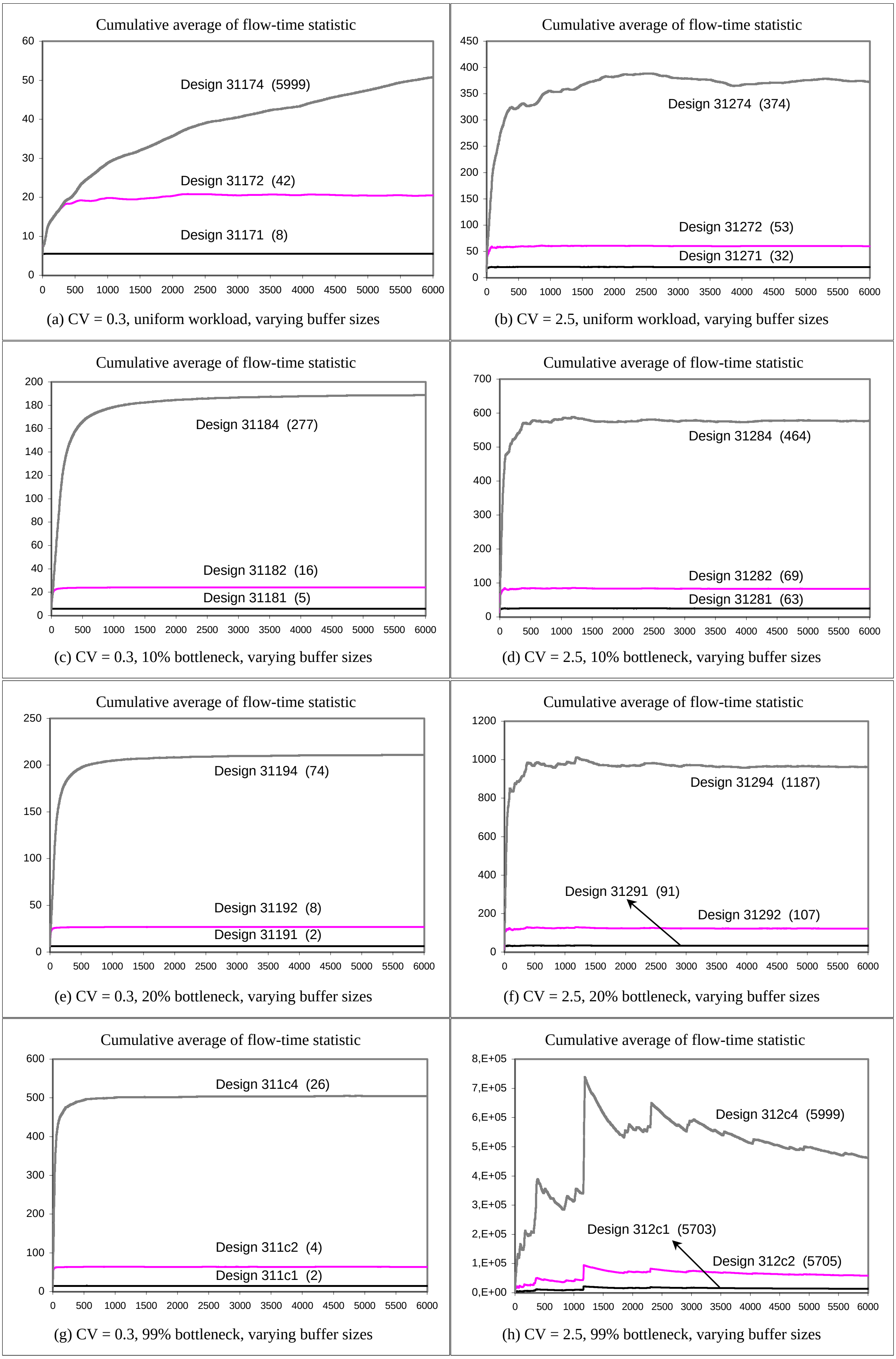


Figure 4.2 Experimental results for 3-stage serial line containing all reliable machines with a total processing time of 2.7 minutes per job.



(a) CV = 0.3, uniform workload, varying buffer sizes

(b) CV = 2.5, uniform workload, varying buffer sizes

(c) CV = 0.3, 10% bottleneck, varying buffer sizes

(d) CV = 2.5, 10% bottleneck, varying buffer sizes

(e) CV = 0.3, 20% bottleneck, varying buffer sizes

(f) CV = 2.5, 20% bottleneck, varying buffer sizes

(g) CV = 0.3, 99% bottleneck, varying buffer sizes

(h) CV = 2.5, 99% bottleneck, varying buffer sizes

Figure 4.3 Experimental results for 3-stage serial line containing all reliable machines with a total processing time of 1.5 minutes per job.

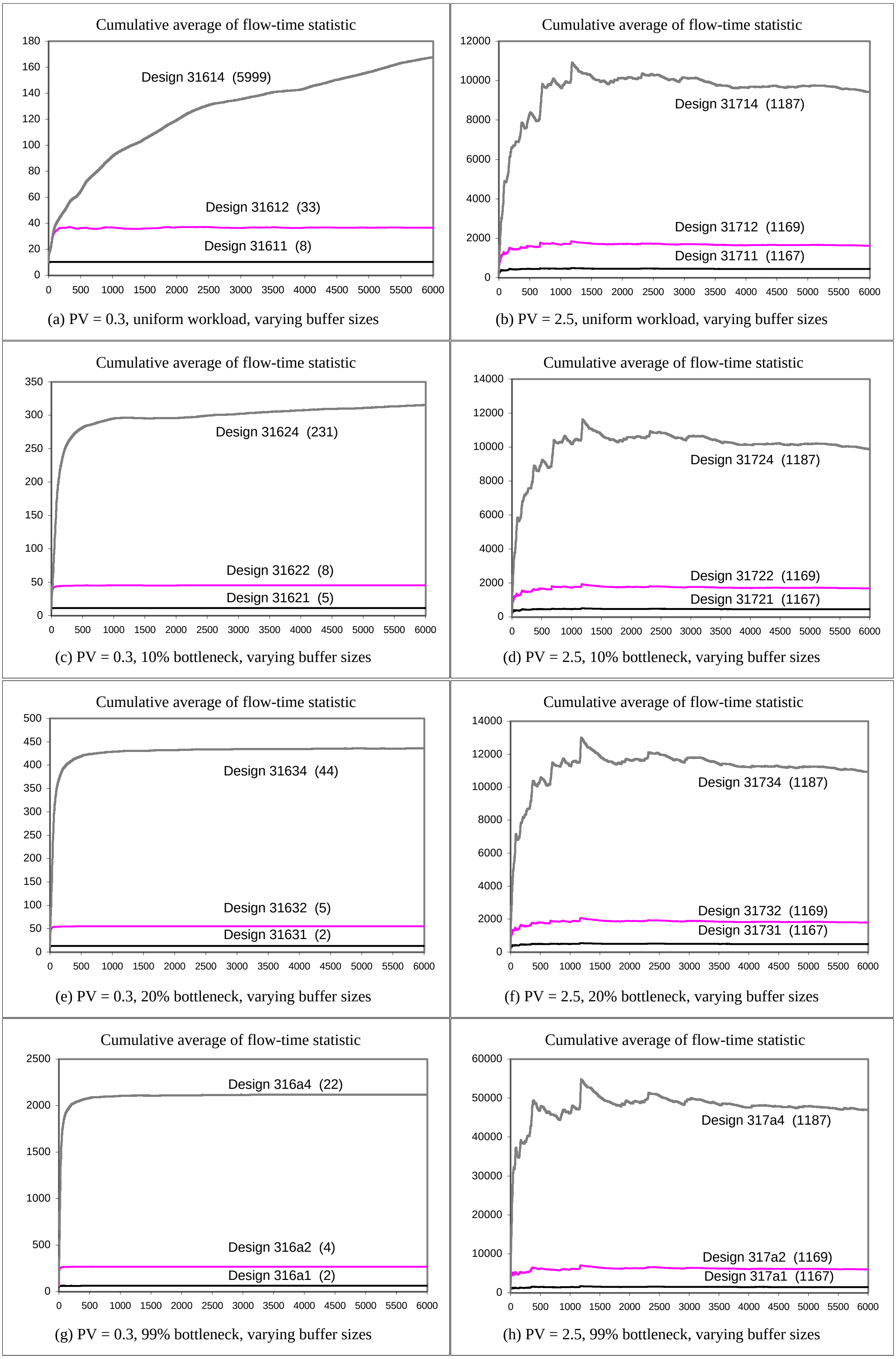
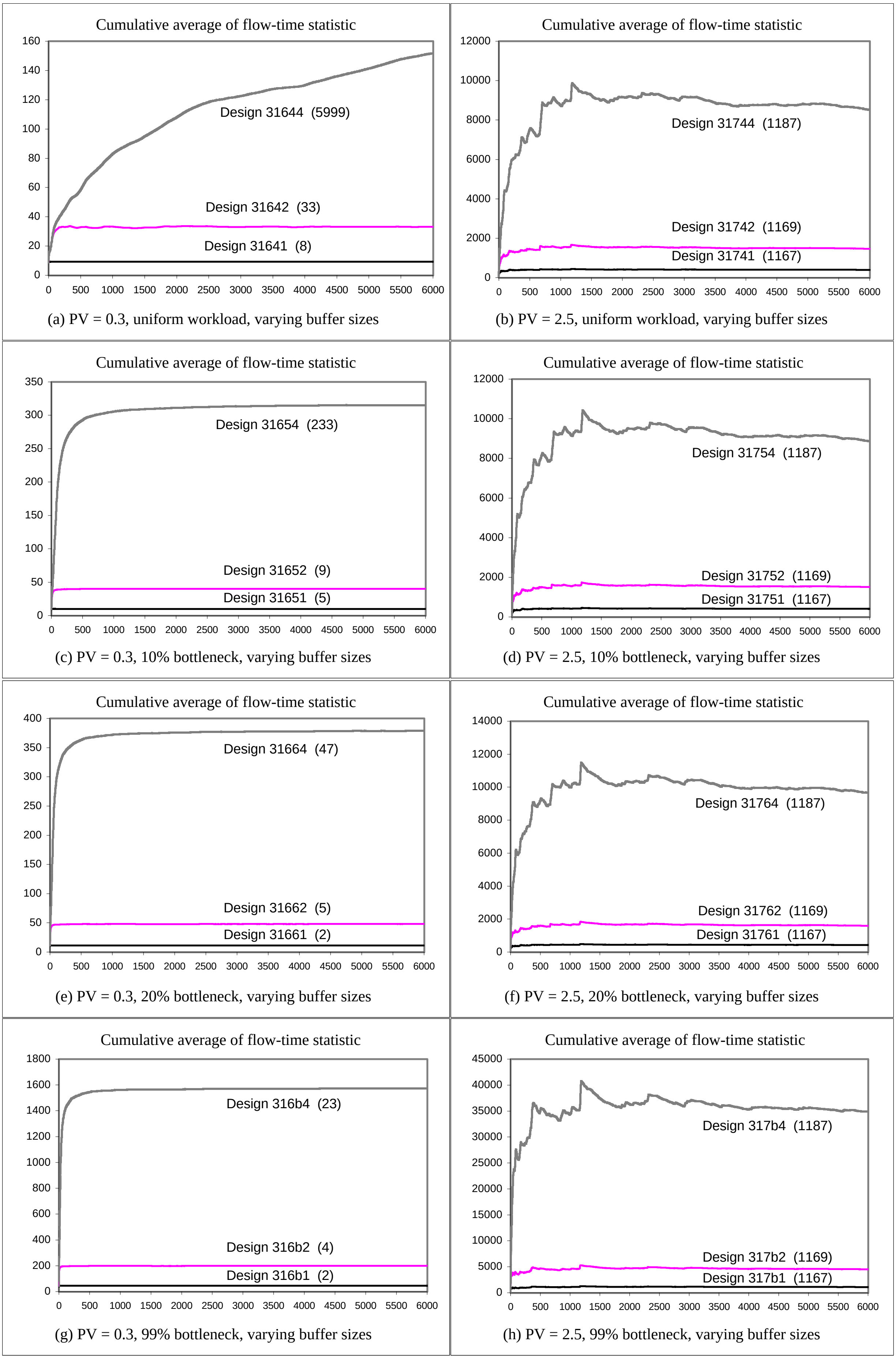


Figure 4.4 Experimental results for 3-stage serial line containing all reliable machines with a total processing time of 3 minutes per job.



(a) PV = 0.3, uniform workload, varying buffer sizes

(b) PV = 2.5, uniform workload, varying buffer sizes

(c) PV = 0.3, 10% bottleneck, varying buffer sizes

(d) PV = 2.5, 10% bottleneck, varying buffer sizes

(e) PV = 0.3, 20% bottleneck, varying buffer sizes

(f) PV = 2.5, 20% bottleneck, varying buffer sizes

(g) PV = 0.3, 99% bottleneck, varying buffer sizes

(h) PV = 2.5, 99% bottleneck, varying buffer sizes

Figure 4.5 Experimental results for 3-stage serial line containing all reliable machines with a total processing time of 2.7 minutes per job.

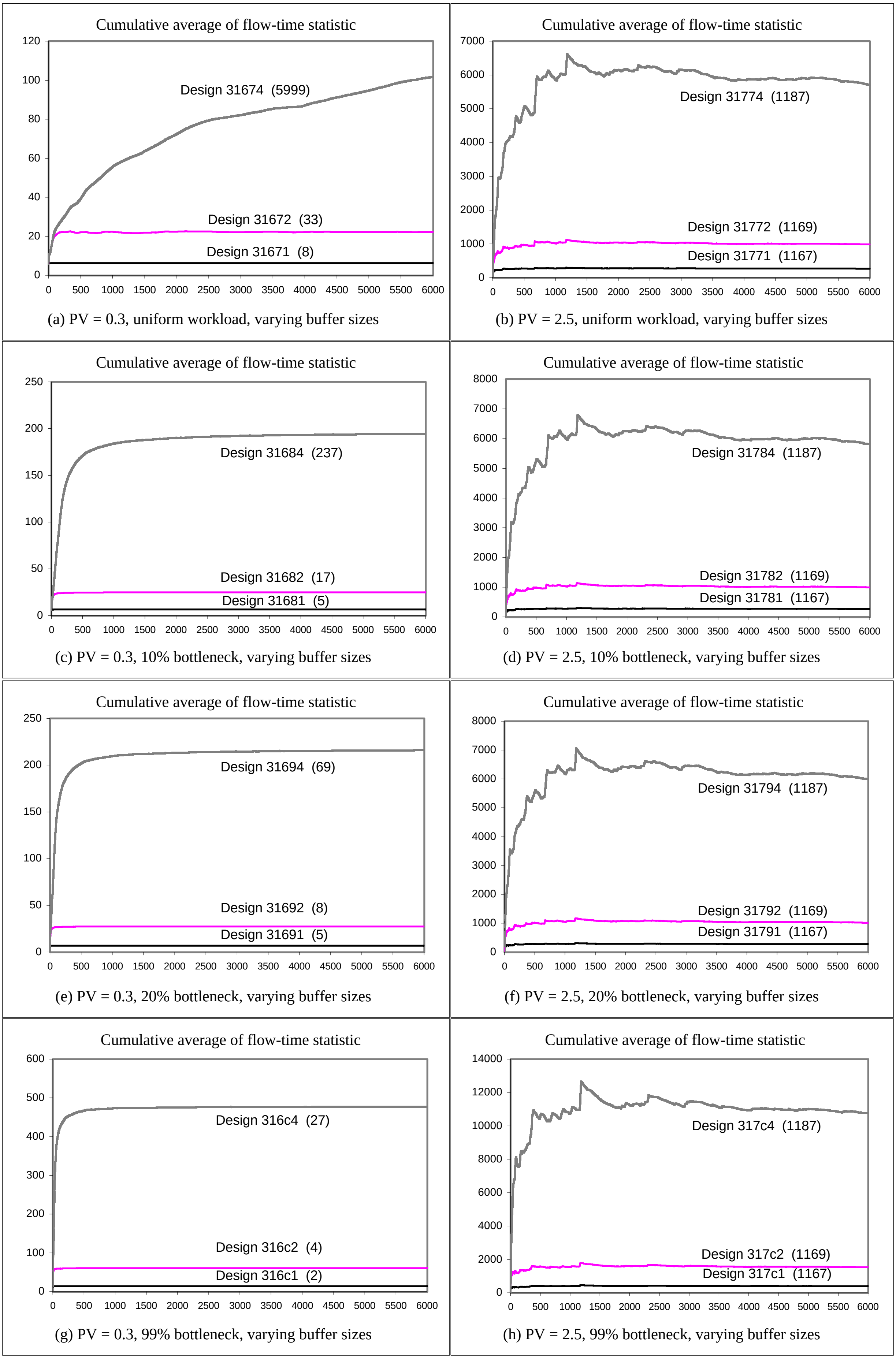
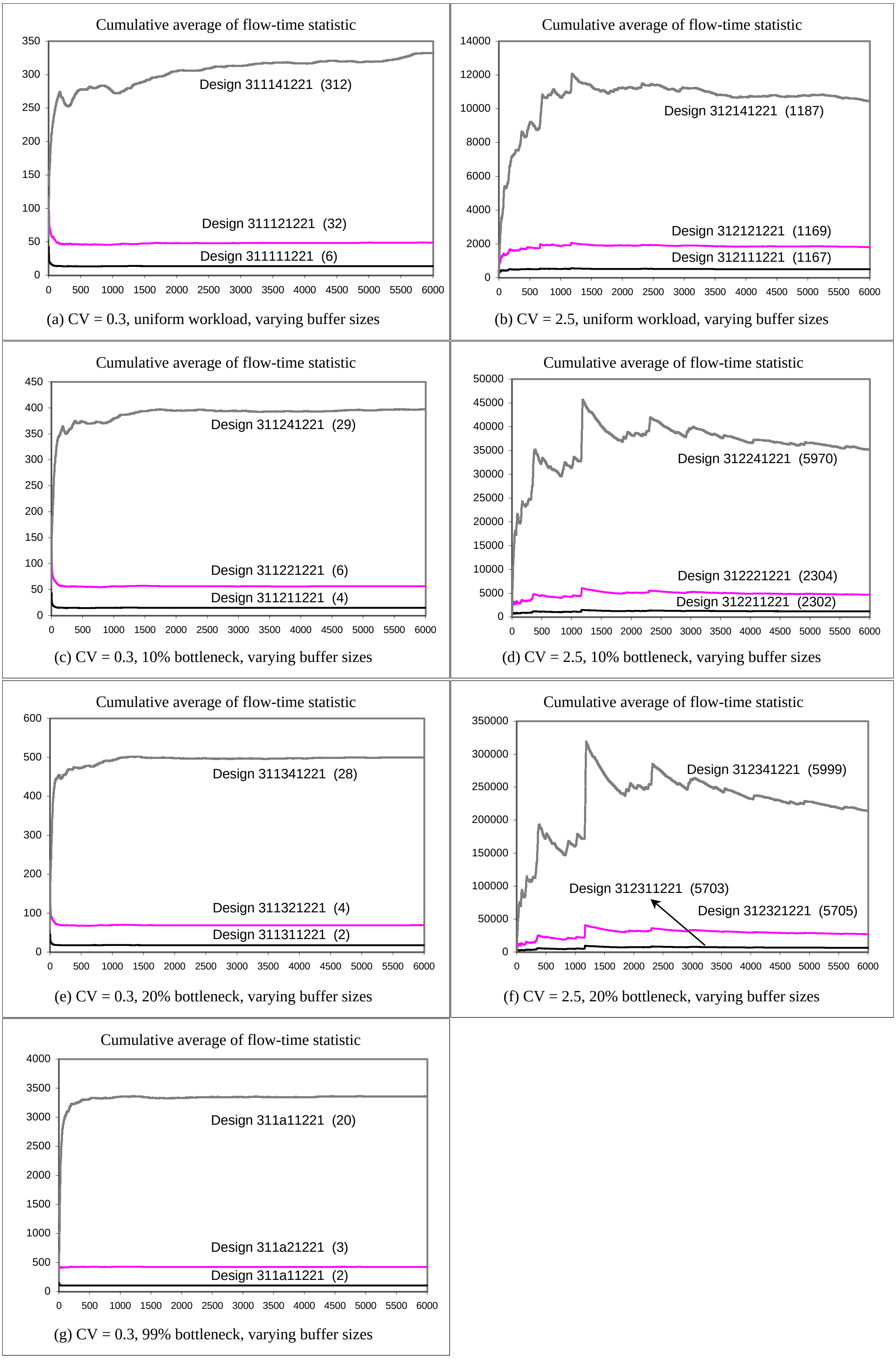


Figure 4.6 Experimental results for 3-stage serial line containing all reliable machines with a total processing time of 1.5 minutes per job.



(a) CV = 0.3, uniform workload, varying buffer sizes

(b) CV = 2.5, uniform workload, varying buffer sizes

(c) CV = 0.3, 10% bottleneck, varying buffer sizes

(d) CV = 2.5, 10% bottleneck, varying buffer sizes

(e) CV = 0.3, 20% bottleneck, varying buffer sizes

(f) CV = 2.5, 20% bottleneck, varying buffer sizes

(g) CV = 0.3, 99% bottleneck, varying buffer sizes

Figure 4.7 Experimental results for 3-stage serial line containing 90% unreliable machines with frequent breakdowns/short repair times and a total processing time of 3 minutes per job.

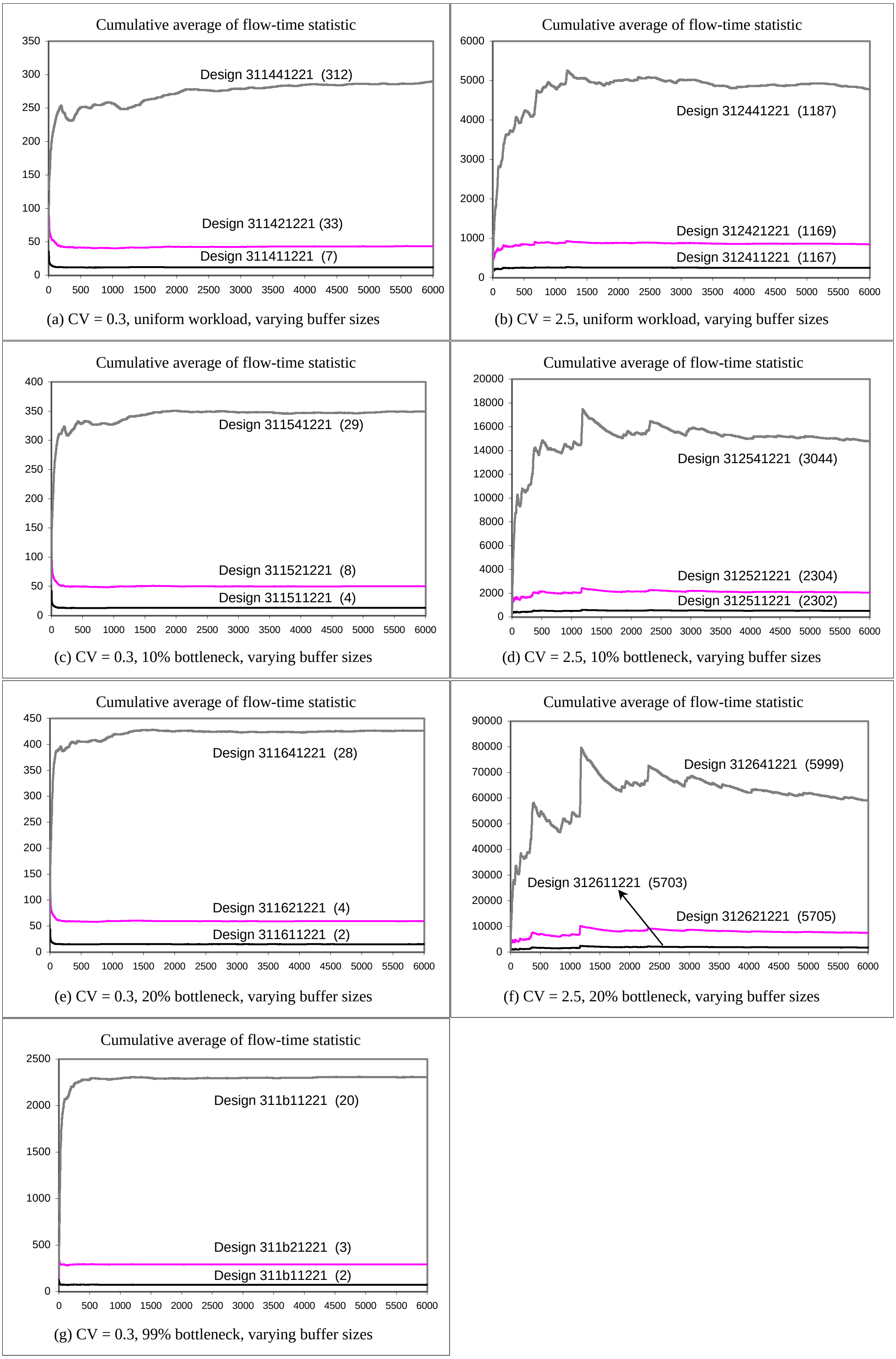


Figure 4.8 Experimental results for 3-stage serial line containing 90% unreliable machines with frequent breakdowns/short repair times and a total processing time of 2.7 minutes per job.

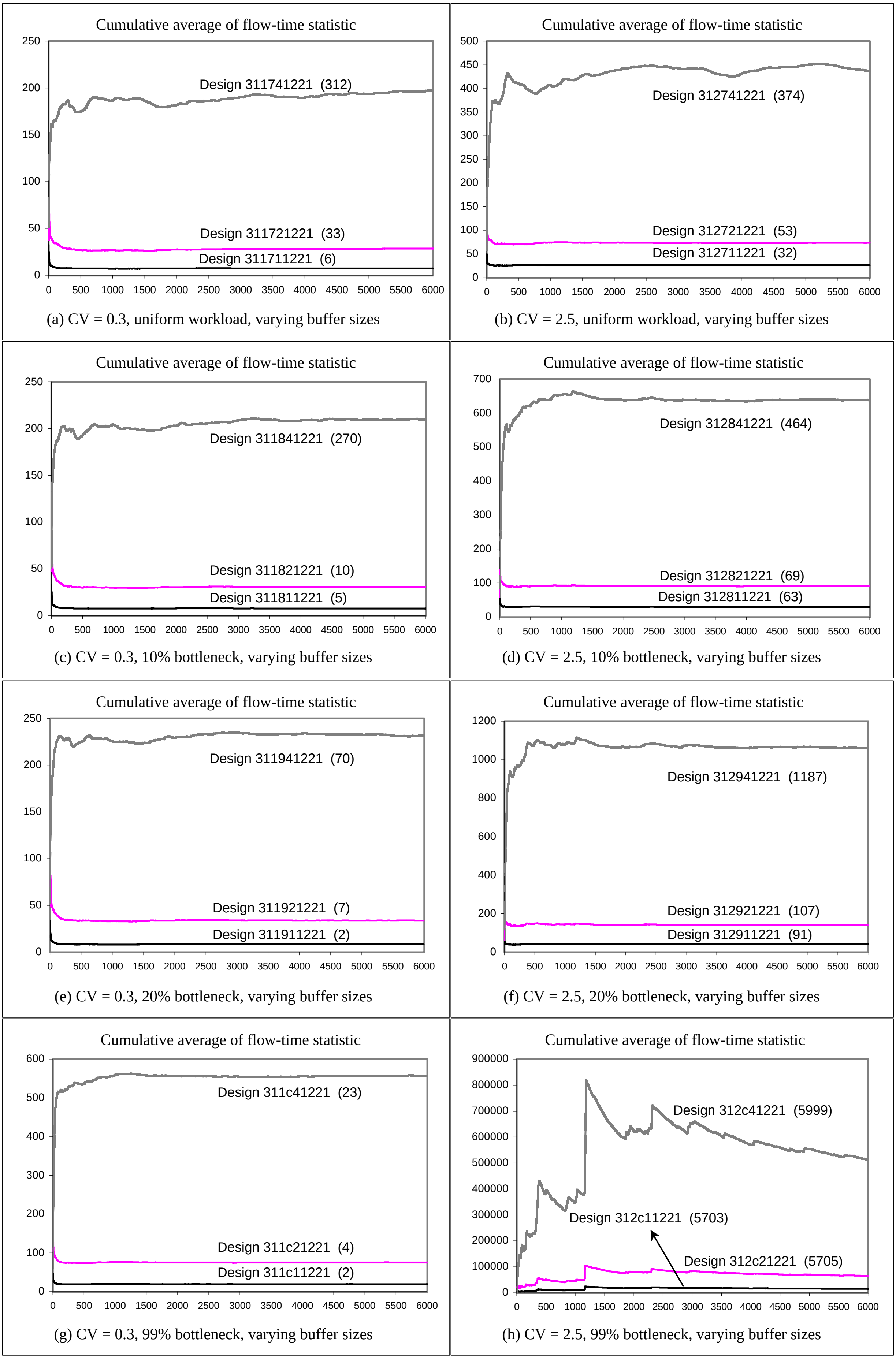


Figure 4.9 Experimental results for 3-stage serial line containing 90% unreliable machines with frequent breakdowns/short repair times and a total processing time of 1.5 minutes per job.

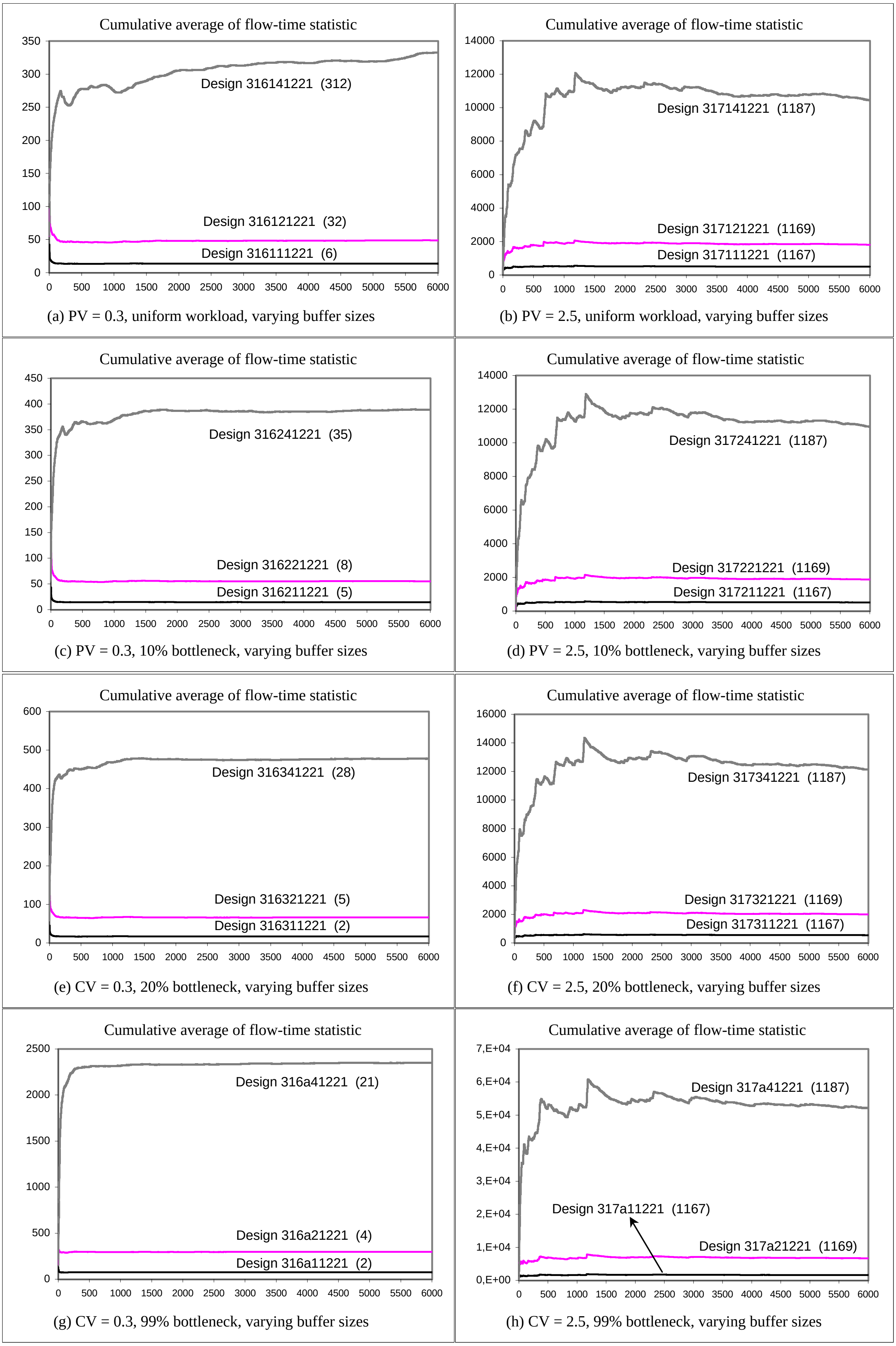


Figure 4.10 Experimental results for 3-stage serial line containing 90% unreliable machines with frequent breakdowns/short repair times and a total processing time of 3 minutes per job.

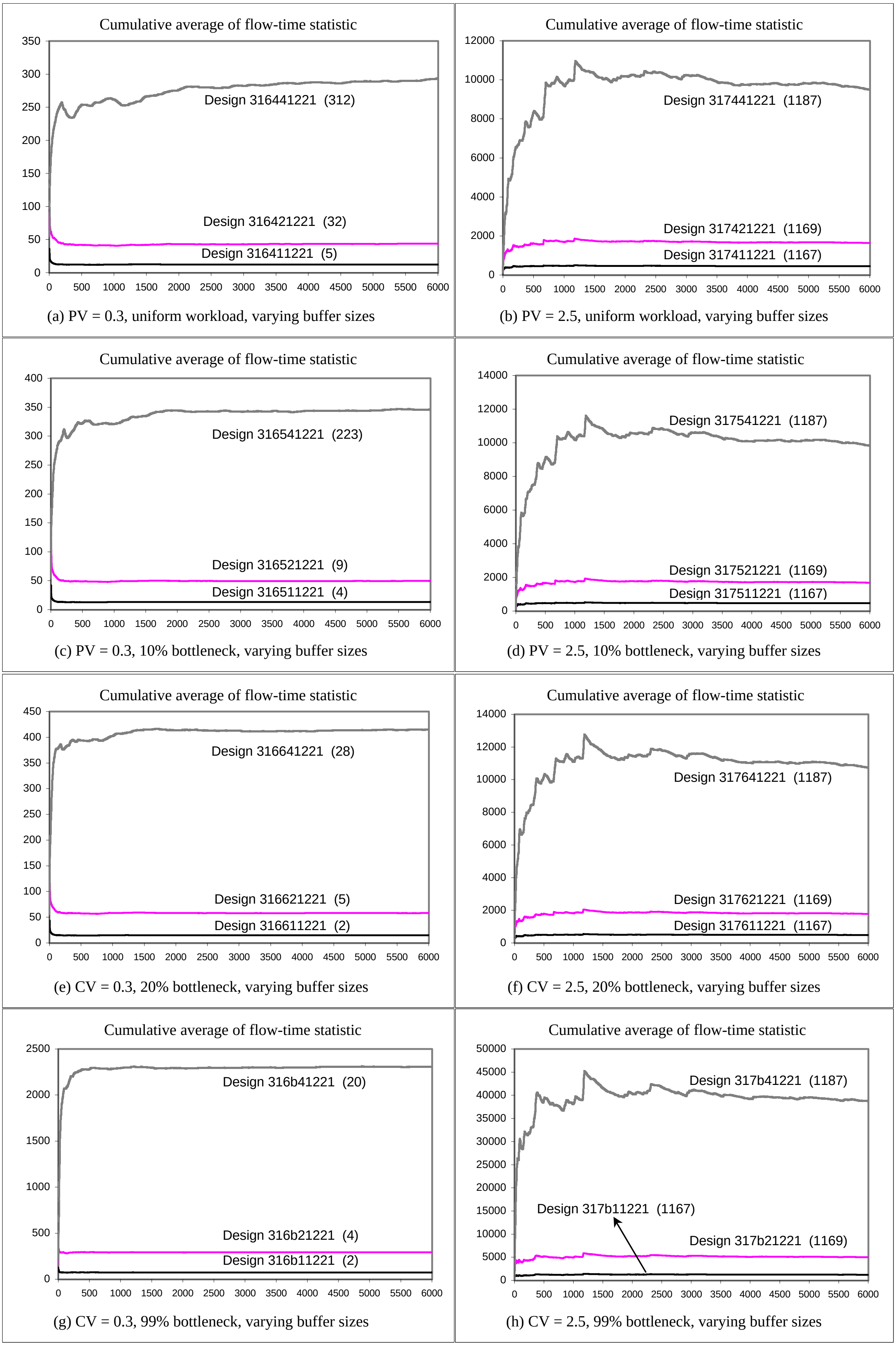
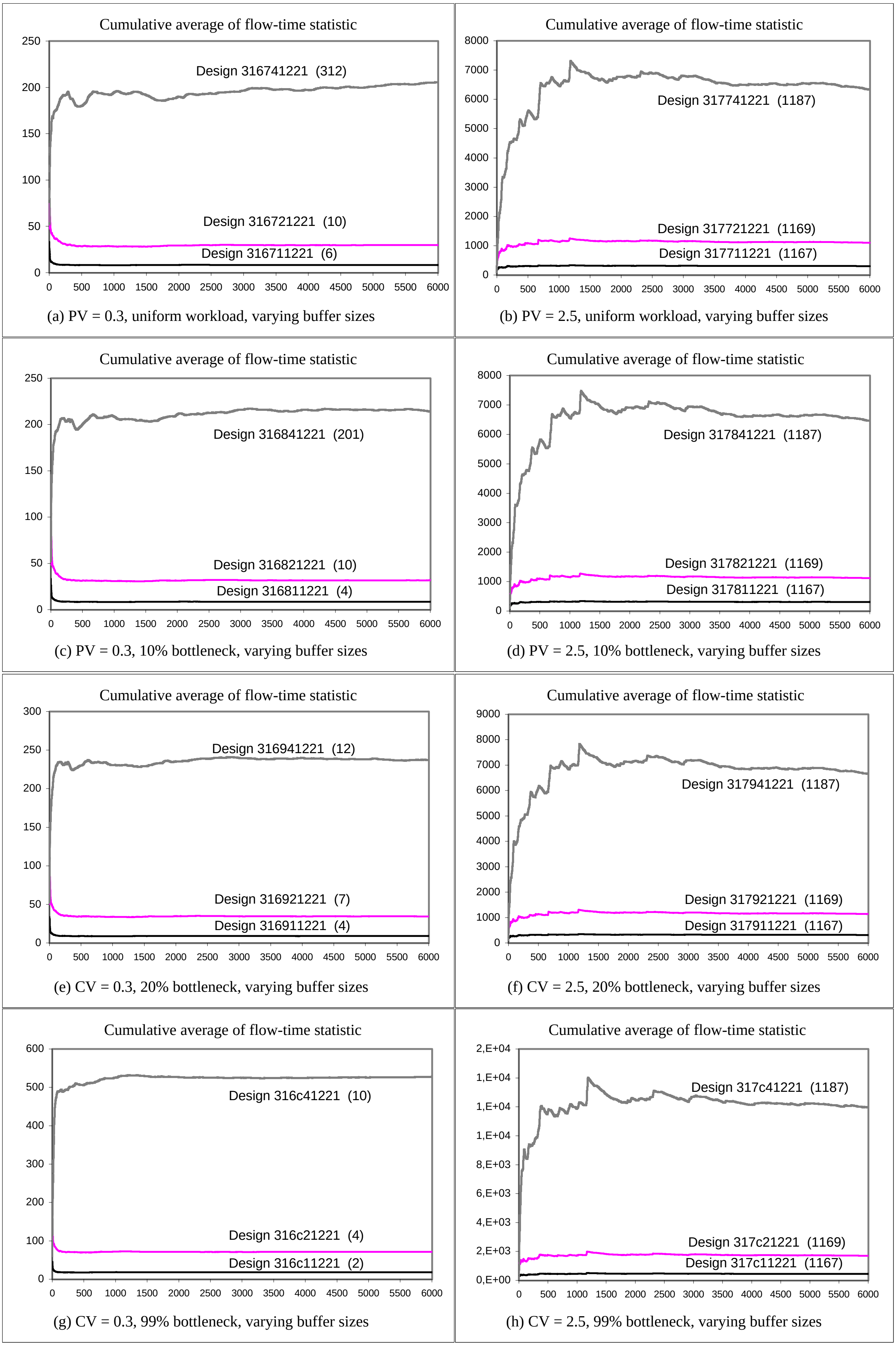


Figure 4.11 Experimental results for 3-stage serial line containing 90% unreliable machines with frequent breakdowns/short repair times and a total processing time of 2.7 minutes per job.



(a) PV = 0.3, uniform workload, varying buffer sizes

(b) PV = 2.5, uniform workload, varying buffer sizes

(c) PV = 0.3, 10% bottleneck, varying buffer sizes

(d) PV = 2.5, 10% bottleneck, varying buffer sizes

(e) CV = 0.3, 20% bottleneck, varying buffer sizes

(f) CV = 2.5, 20% bottleneck, varying buffer sizes

(g) CV = 0.3, 99% bottleneck, varying buffer sizes

(h) CV = 2.5, 99% bottleneck, varying buffer sizes

Figure 4.12 Experimental results for 3-stage serial line containing 90% unreliable machines with frequent breakdowns/short repair times and a total processing time of 1.5 minutes per job.

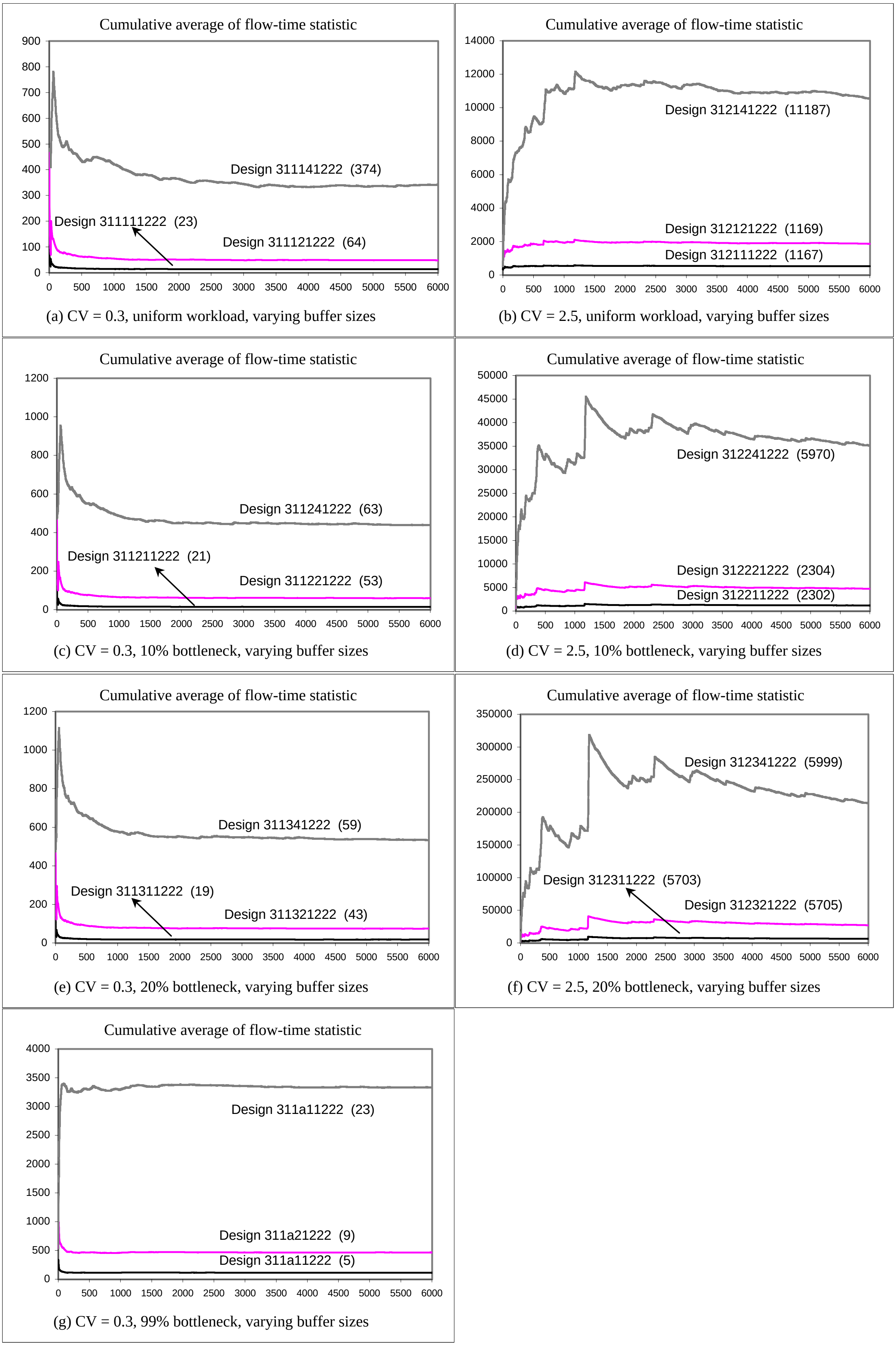
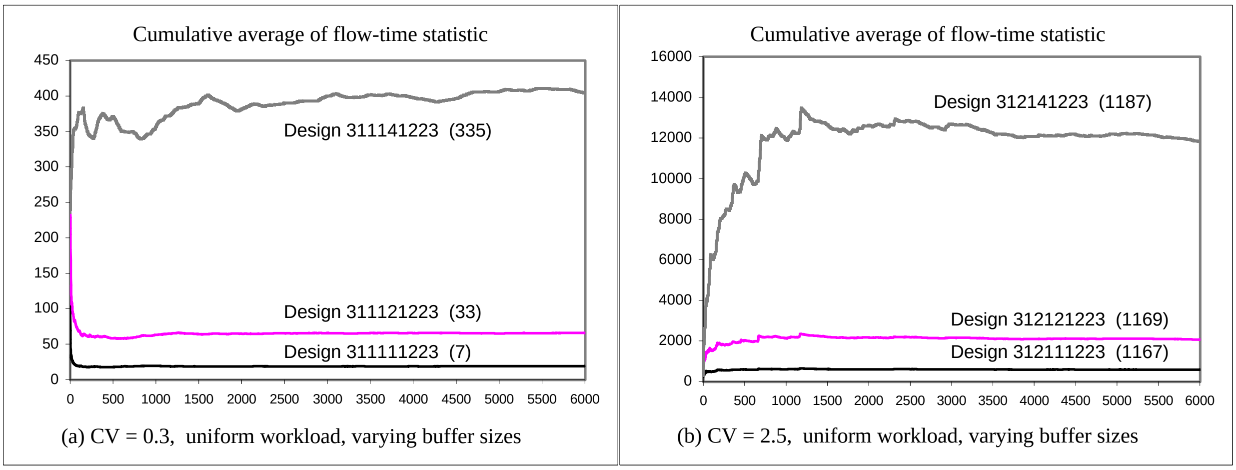


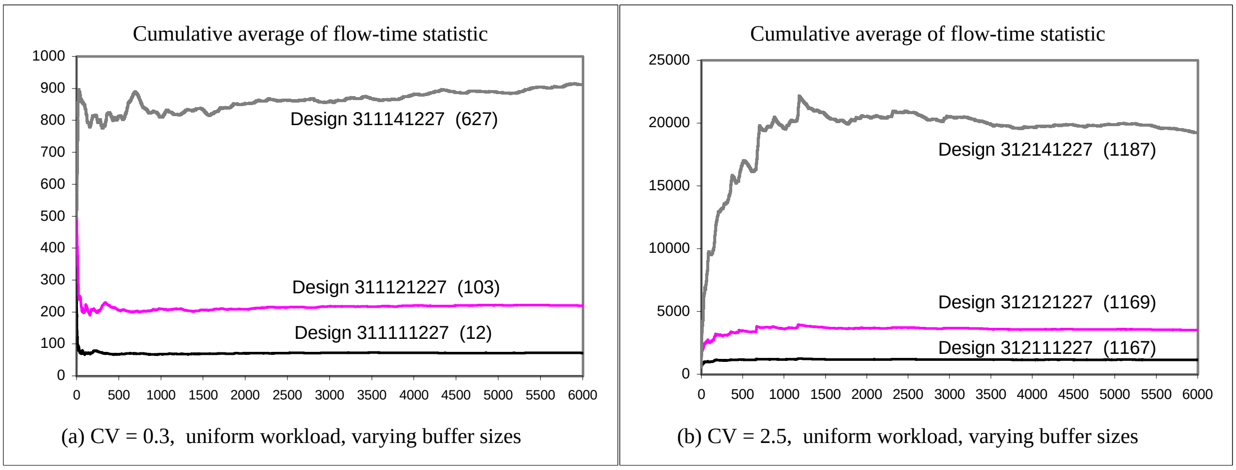
Figure 4.13 Experimental results for 3-stage serial line containing 90% unreliable machines with rare breakdowns/long repair times and a total processing time of 3 minutes per job.



(a) CV = 0.3, uniform workload, varying buffer sizes

(b) CV = 2.5, uniform workload, varying buffer sizes

Figure 4.14 Experimental results for 3-stage serial line containing 80% unreliable machines with frequent breakdowns/short repair times and a total processing time of 3 minutes per job.



(a) CV = 0.3, uniform workload, varying buffer sizes

(b) CV = 2.5, uniform workload, varying buffer sizes

Figure 4.15 Experimental results for 3-stage serial line containing 50% unreliable machines with frequent breakdowns/short repair times and a total processing time of 3 minutes per job.

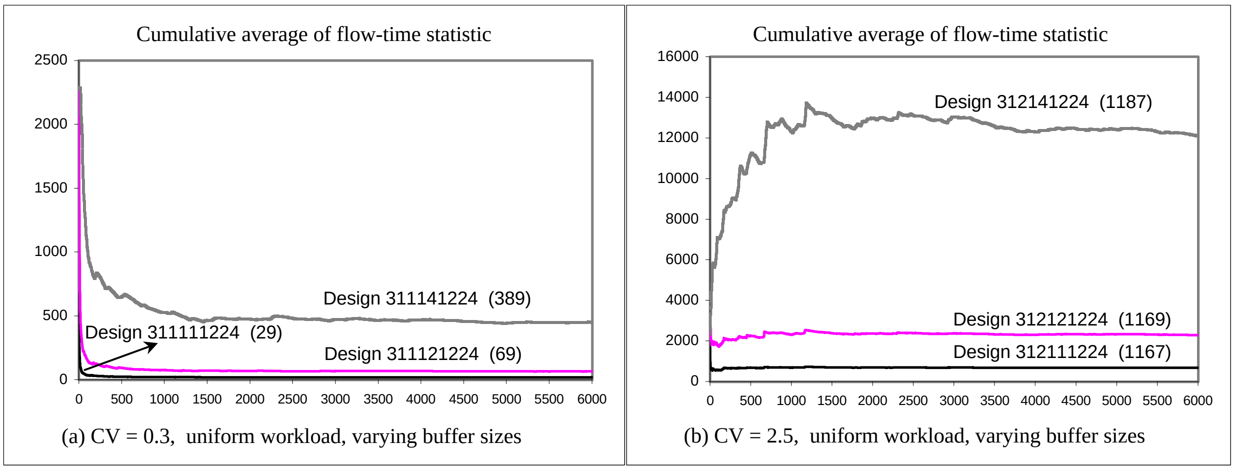


Figure 4.16 Experimental results for 3-stage serial line containing 80% unreliable machines with rare breakdowns/long repair times and a total processing time of 3 minutes per job.

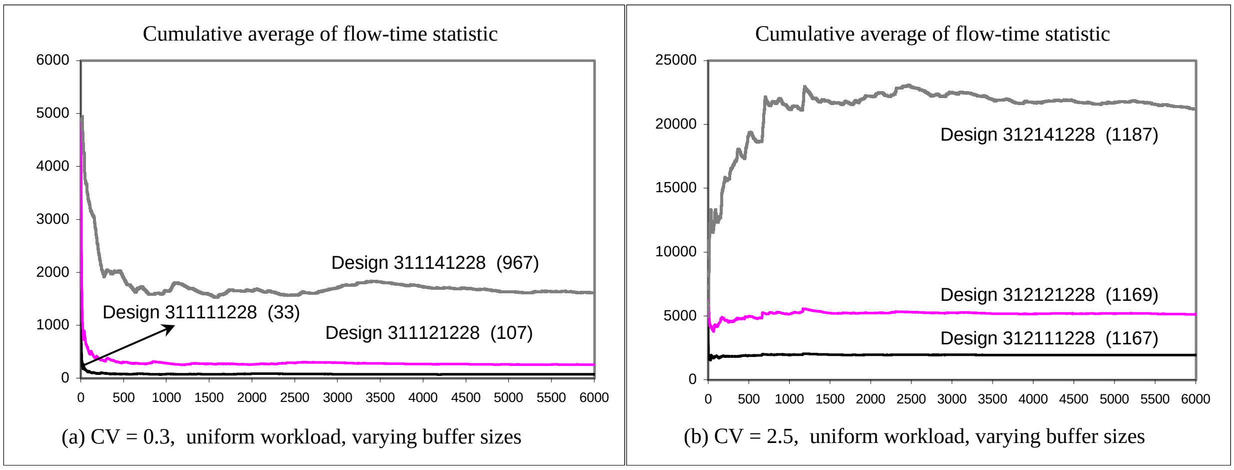
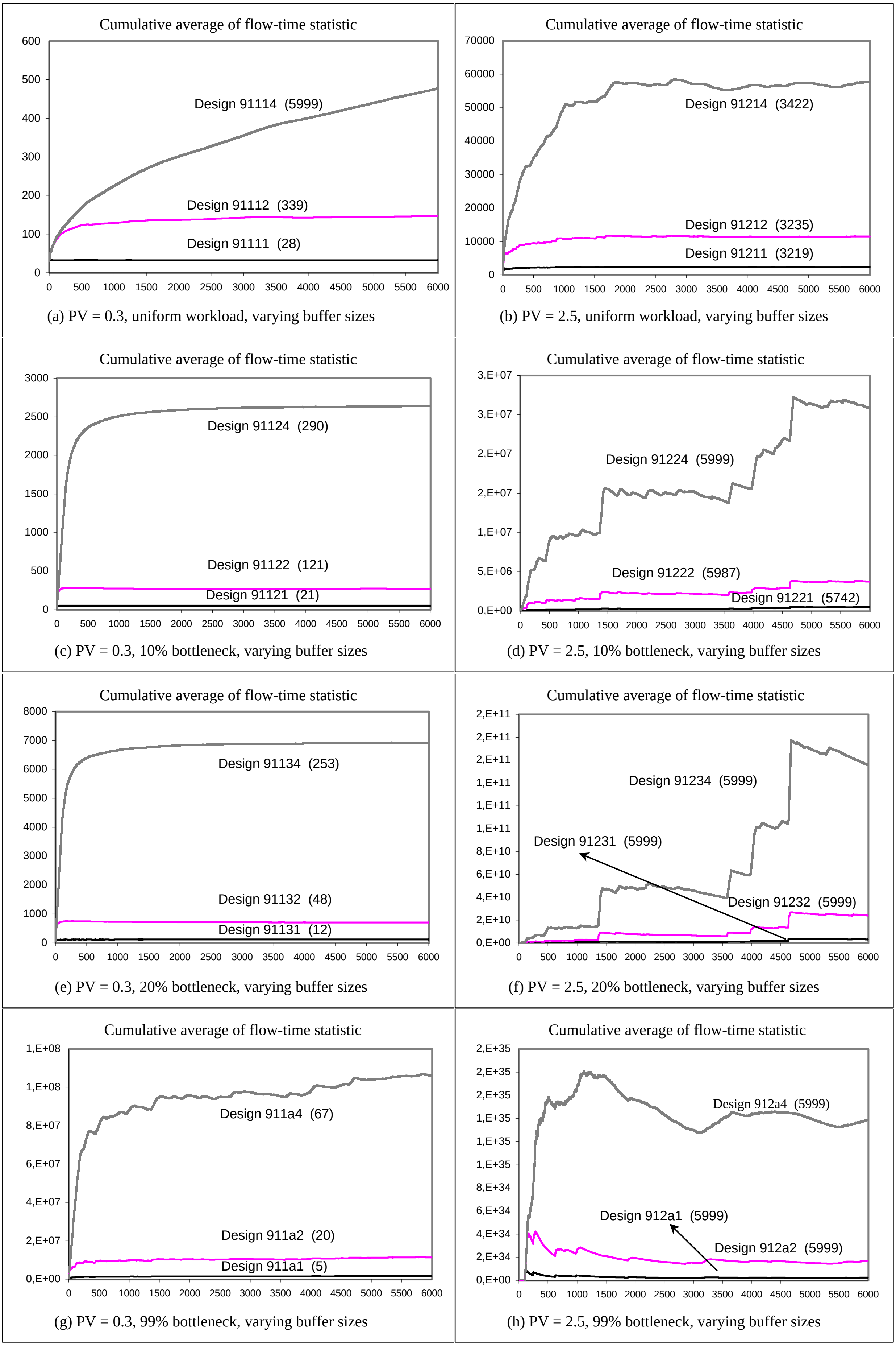


Figure 4.17 Experimental results for 3-stage serial line containing 50% unreliable machines with rare breakdowns/long repair times and a total processing time of 3 minutes per job.



(a) PV = 0.3, uniform workload, varying buffer sizes

(b) PV = 2.5, uniform workload, varying buffer sizes

(c) PV = 0.3, 10% bottleneck, varying buffer sizes

(d) PV = 2.5, 10% bottleneck, varying buffer sizes

(e) PV = 0.3, 20% bottleneck, varying buffer sizes

(f) PV = 2.5, 20% bottleneck, varying buffer sizes

(g) PV = 0.3, 99% bottleneck, varying buffer sizes

(h) PV = 2.5, 99% bottleneck, varying buffer sizes

Figure 4.18 Experimental results for 9-stage serial line containing all reliable machines with a total processing time of 9 minutes per job.

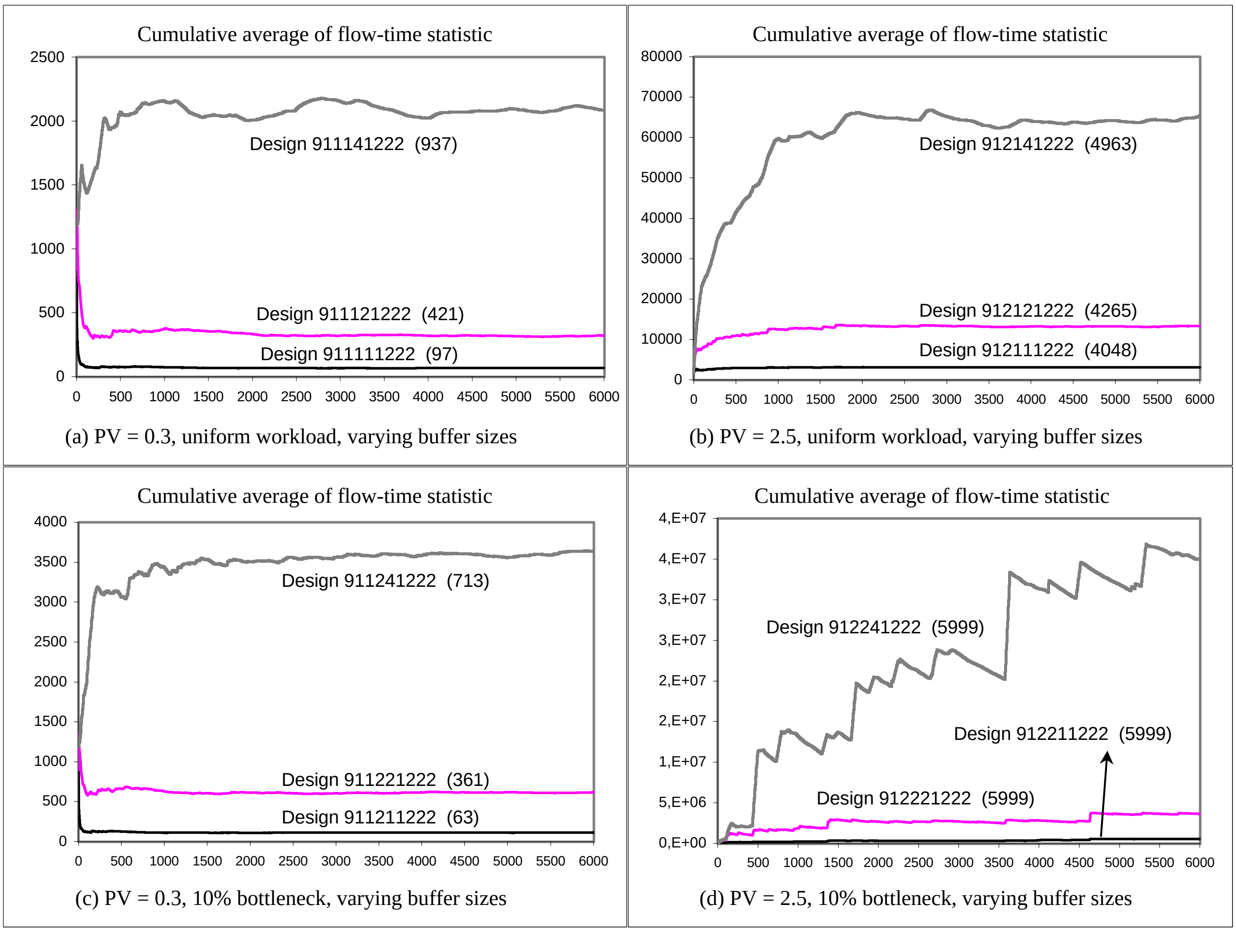


Figure 4.19 Experimental results for 9-stage serial line containing 90% unreliable machines with rare breakdowns/long repair times and a total processing time of 9 minutes per job.

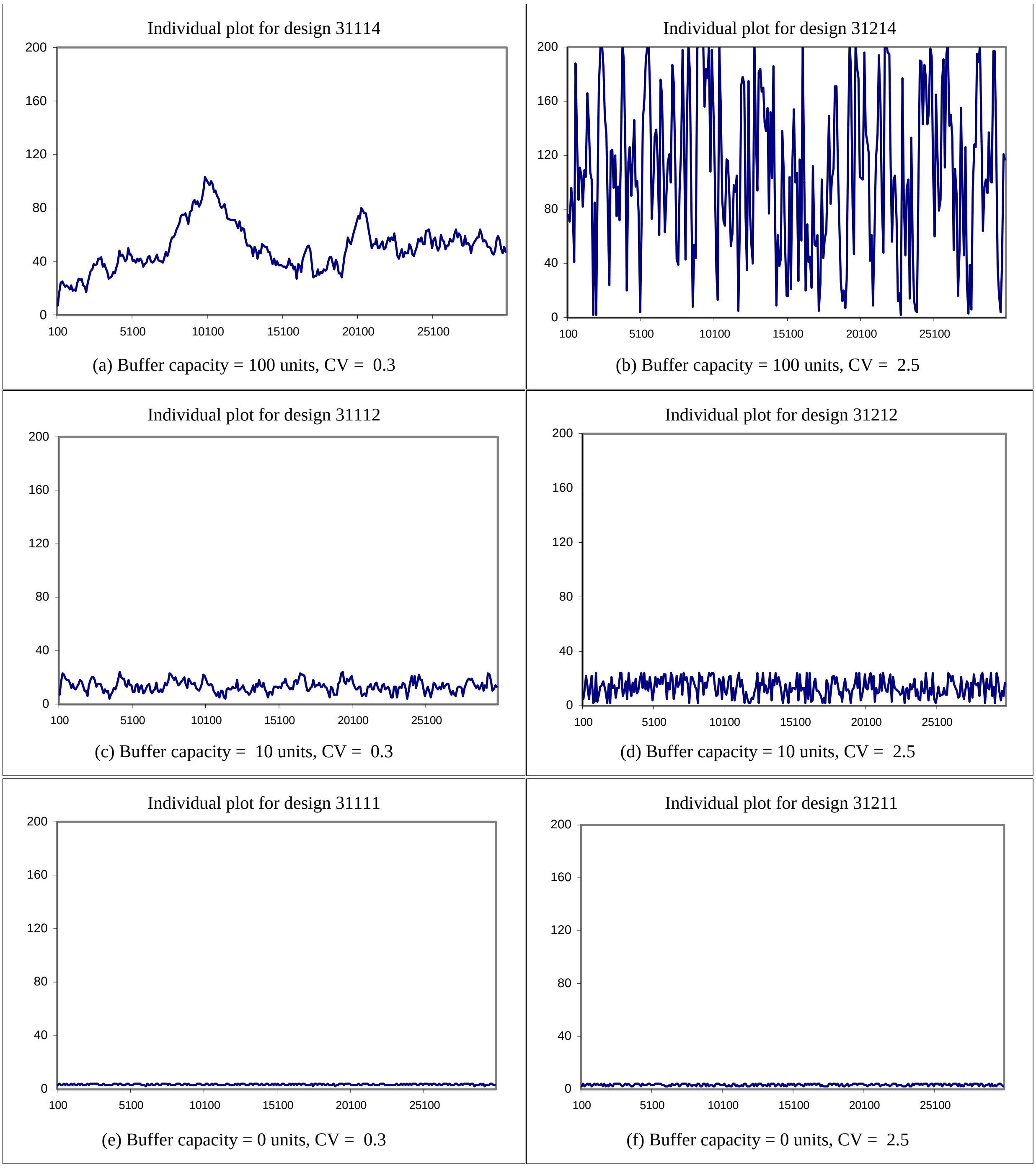


Figure 4.20 Number of jobs in system for 3-stage reliable serial line.

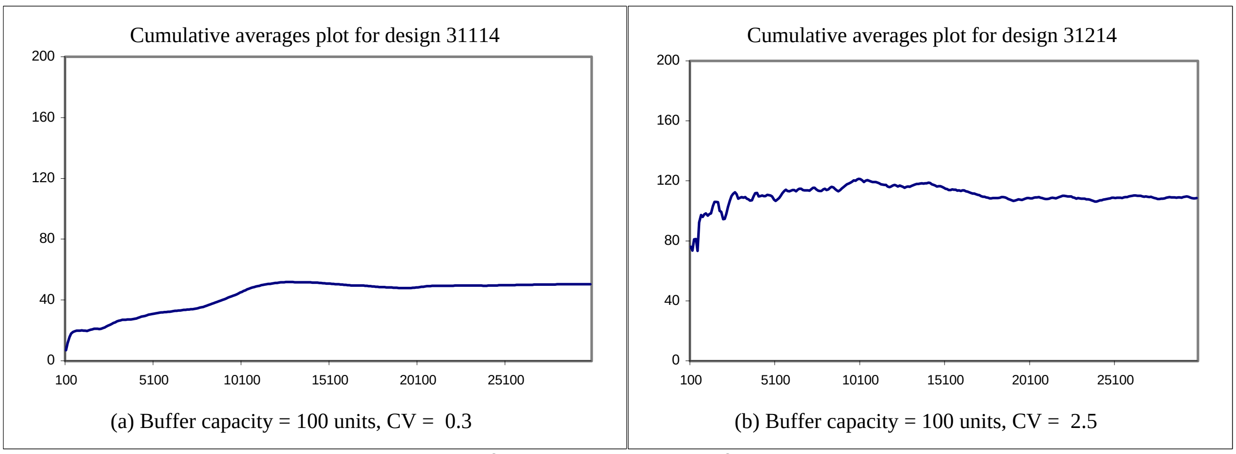


Figure 4.21 Cumulative averages plot of number in system for 3-stage serial line

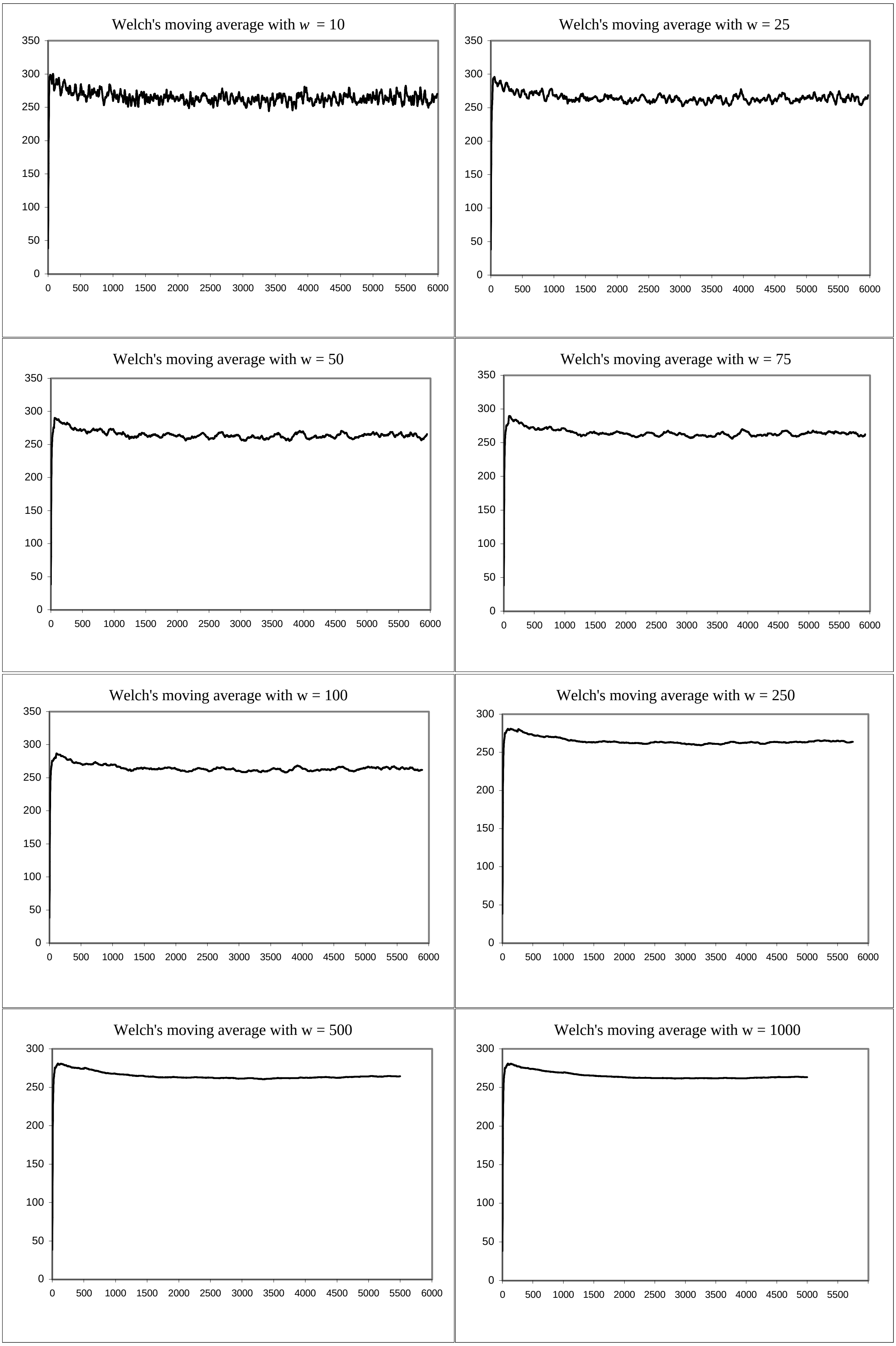


Figure 4.22 Determining the length of transient period using Welch's technique for design 91122.

5 RESULTS FOR JOB-SHOP PRODUCTION SYSTEMS

This chapter is an extension of the research presented in Chapter 4. Specifically, it includes the results of the investigation for factors affecting the length of the transient period in job-shop production systems. Similar to the previous chapter, we first outline the structural framework for the presentation of outputs, which slightly differ from that of serial line results. Then, we continue in each section with presenting the effects of different system factors on the length of the transient period.

The structural framework in the presentation of outputs

The figures in this section can be viewed as a 3 × 2 matrix format. That is, there are 3 rows and 2 columns in *most* of the figures, with each row-column intersection containing a small figure. Where appropriate, we followed the same structure used for presenting the serial line results. However, there still exist slight differences. To ease the job of following the relationships between different designs, we structure the figures in the following manner.

The first column of each figure is composed of the designs with low variable (either CV or PV) processing times, whereas the second column is composed of the designs with highly variable (either CV or PV) processing times. Moreover, the rows within a figure present the designs with different workload distributions. The first row is composed of the designs where the total workload is distributed uniformly among machines. The second row is composed of designs that include a 5% bottleneck station, i.e., the total processing time is distributed to the machines in such a manner that a randomly selected machine, say machine #2 in 3-machine systems, takes 5% of the processing times of the other machines. Similarly, the third row is composed of designs that involve a 10% bottleneck machine. Furthermore, each of the small figures in a row-column intersection includes two separate lines. These lines correspond to the cumulative averages plot of flow-time statistic for the designs that differ only in load levels. The name of the design is shown next to the corresponding plots. (Note that the complete list of design names with their parameter levels is presented in Table C.2 of Appendix C.) Additionally, the numbers in parentheses next to the design names are the truncation points suggested by the MSER-5 heuristic. The *x*-axis of each figure shows the number of observations in the sequence, whereas the *y*-axis shows the flow-time of parts in minutes.

The following example clarifies the discussion. The two lines in Figure 5.1 (a) correspond to designs `31111` and `31171`. The truncation points suggested by MSER-5 heuristic for these designs are 25 and 13, respectively. Designs in Figures 5.1 (a) (i.e., `31111` and `31171`) and 5.1 (b) (i.e., `31211` and `31271`) are exactly the same designs with a load level of 80% and 50%, respectively, except the variability. The former designs have a CV of 0.3 whereas the latter ones have a CV of 1.0. The figures in other rows are interpreted similarly. Moreover, the designs in Figures 5.1 (a) and 5.1 (c) differ only on the distribution of the total processing time to the machines. The processing times per job of the machines in Figure 5.1 (a) are 1-1-1 minutes, whereas it is 0.95-1.1-0.95 (the order of sequence is random) minutes for the machines in Figure 5.1(c) (5% bottleneck machine). The depth of the bottleneck is further increased to 10% in Figure 5.1 (e) resulting in a workload distribution of 0.9-1.2-0.9, respectively. The same pattern is followed in the second column, as well.

Most of the figures in this section share the above discussed structure. However, since we did not perform a full factorial experimentation due to a recognized pattern, some of the figures are left incomplete in the sense that they do not have 3 × 2 matrix structure. Nevertheless, the basic structure of these figures also complies with the above discussion.

5.1 The effect of processing time variability

As we did for the serial line systems, we distinguish between two alternative measures of variability, namely, coefficient of variation (CV) and variance (PV). The experimental results for the effect of variability measured by CV on the 3-machine reliable job-shop are shown in Figure 5.1 and results for the unreliable case are shown in Figures 5.3 and 5.5. The effect of variability measured by PV for the same system considerations are shown in Figures 5.2 for the reliable case, and in Figure 5.4 and 5.6 for the unreliable case. Experimentally, it is found that *variability of the processing times of the machines has significant negative effect on the length of the transient period* (i.e., transient period increases as variability increases.) The explanation discussed in Section 4.2 about the effect variability on the length of the transient period for serial line systems is also valid for the job-shop systems, as well. That is, as the variability of the processing times of machines increases, the overall system variability increases, too, which in turn results with longer transient periods.

Consider, for example, Figures 5.1 (a) and 5.1 (b) for low variable and highly variable job-shop systems, respectively. According to cumulative averages plot, the highly loaded system in the low variable case, i.e., design `31111`, reaches steady-state at the 100th observation whereas the corresponding system in the highly variable case, i.e., design `31211`, reaches steady-state at the 250th observation. MSER-5 heuristic also complies with the cumulative averages findings by suggesting a truncation amount of 25 and 43 observations for designs `31111` and `31211`, respectively. The results for low loaded system also confirm this observation. Similar observations about the effect of variability can be made for the rest of Figure 5.1. The same observations are also made for the variability measured by PV (see Figure 5.2), for unreliable systems (see Figures 5.3, 5.4, 5.5, and 5.6), and for larger systems sizes (see Figures 5.7, 5.8, and 5.9).

5.2 The effect of system size

As was found for serial lines systems, our observations show that *system size* measured in terms of number of machines in the system *has a significant negative effect on the length of the transient period*. As can be seen from the comparison of systems with all reliable machines, increasing number of machines in the system increases the length of the transient period (each individual plot in Figure 5.1 (a) is compared to that of Figure 5.7(a), and 5.1 (b) to 5.7 (b).) In case of considering the highly loaded systems, it is seen that the transient period increases as the system size increase. For example, design `31111` in Figure 5.1 (a) reaches steady-state at the 25th observation whereas its counterpart, i.e., design `91111`, in Figure 5.7 (a) reaches steady-state at the 28th observation. The systems including low load levels also confirm to the fact that as the number of machines in the system increases, the length of the transient period also increases. This is mainly due to more coupling in larger systems. The stove analogy described in Section 4.3 can be used for job-shop systems, as well. Similar results can be seen for the systems with all unreliable machines (compare Figure 5.3 to Figure 5. 8, and Figure 5.5 to Figure 5.9.)

5.3 The effect of distribution of system load

The results of simulation experiments also indicated that the distribution of system load (or *the bottleneck* issue) *has significant negative effect on the length of the transient period*. This behavior is observed for both constant CV and constant PV cases. This results is a little bit different than the serial line results as it was explained in detail.

Consider, for instance, the highly loaded, uniform, 3-machine reliable job-shop, i.e., design `31171` in Figure 5.1 (a), which reaches steady state at the 13th observation according to the MSER-5 heuristic. The 5% and 10% bottleneck versions of this design, i.e., designs `311y1` and `31181` in Figures 5.1 (c) and 5.1 (e) reaches steady-state at the 15th and 36th observation, respectively.

The explanation of this kind of behavior in job-shop system regardless of the type of variability is as follows. Firstly, it is important to recognize that *Observation* 1 which is stated in Section 4.4 (i.e., "as the depth of bottleneck increases the size of the system decreases") is also valid for the job-shop systems. However, two basic differences between job-shop and serial line systems are:

1. The type of flow in two systems differs. That is, there is single fixed flow sequence of jobs in serial lines that regulates the system. However, the jobs follow a random processing sequence in a job-shop system, which results with additional variability.
2. There is a structural difference between two systems. The collection of time in system statistic of a job in serial line systems is started as the job begins being processed in the first machine. However, the same statistic in a job-shop system is started as the job first arrives the system (notice that a job may not start processing immediately it arrives to a system in job-shop). This difference causes a significant change in the congestion levels of two systems. Clearly, job-shop systems is more congested than serial line systems. Congestion causes a job-shop system that contains bottleneck to become unstable. However, a similar serial line system is limited with the buffer sizes. Hence, the significant change in the congestion level causes significant degree of variability.

This can be observed for the examples discussed above. The mean and variance of time in system statistic for design `31171` is 12.465 and 19.67, respectively. The mean for designs `311y1` and `31181` has slightly increase to 12.749 and 13.587, whereas the variances increased up to 23.12 and 35.23, respectively.

In summary, by allowing bottleneck machines in a job-shop we are in essence disturbing the balance between machines, which in turn causes an introduction of additional variability. A reduction in system size might also occur, which suggest a decrease in the transient period. However, the negative effect of variability dominates the positive effect of system size. The same kind of behavior is observed for unreliable systems (see Figures 5.3 thorough 5.6) and larger systems (see Figures 5.7 thorough 5.9).

5.4 The effect of system load level

System load level as also measured by the average utilization of system, which is adjusted by the rate of the arrival process. The interarrival time of jobs for a 50% utilized low variable system is set to 5.65 minutes. To increase the average utilization of this system to 80% the interarrival time of jobs is decreased down to 3.55 minutes. This decrease in interarrival time will lead to frequent arrival of jobs, which in turn will give less opportunity to machine for being idle.

The results indicate that *increasing the load level* in job-shop systems *causes a significant increase in the length of the transient period*. Similar to the effect of distribution of system load, the same kind of behavior is observed for the effect of load level in job-shop experiments for constant CV and constant PV cases, which was also differentiated in serial line experiments.

Each individual plot within a row-column intersection of Figures 5.1 thorough 5.9 should be compared to each other to see the effect of different load levels. For instance, the 80% utilized 3-machine reliable system with low variable processing times, i.e., design `31111` in Figure 5.1 (a) should be compared to design `31171` in Figure 5.1 (a), which is exactly the same design except the decrease in load level (from 80% to 50%). MSER-5 heuristic suggests truncating 25 and 13 observations for designs `31111` and `31171`, respectively. Cumulative averages plots also confirm MSER-5 findings. Similar observations are made for highly variable designs (see right column of Figure 5.1), designs containing bottleneck (see second and third row of Figure 5.1), unreliable designs (see Figures 5.3 through 5.6), and for larger systems (see Figures 5.7 thorough 5.9).

This is a counterintuitive result, because as we increase the load level we are in essence reducing the coupling in system. In job-shop systems, coupling occurs only in the form of starvation, because as long as a machine finishes it's processing on a job it can dispose it to the system regardless of anything. Since increased load level will increase the number of jobs residing in system, this will in turn give less opportunity for the starvation of machines. Hence, this will result with less coupling which suggests a shorter transient period.

However, if investigated carefully one would observe that increasing the load level of a system also increases the variability of the system. This is the main reason for the increase in the length of the transient period.

Let us define the lead time of a job in a machine be the total time a job spends for being processed in a machine plus the time it spends for waiting the machine to be available. Hence, lead time of a job is always greater than or equal to its processing time. By this definition, it can easily be seen that total lead time (i.e., the sum of lead times in each machine) is equal to the flow-time of the job in the system. The congestion level of a low loaded system is lower than that of high loaded system. Therefore, mean total lead time is clearly greater for a highly loaded system than that of low loaded one. Since common random numbers is used, mean processing times in each machine for each system will exactly be equal to each other. However, the mean lead time and its variance in each machine will differ for each system. Tables 5.1 presents the mean, variance and the coefficient of variation of both lead time and processing time in each machine for both low loaded and highly loaded 3-machine reliable job-shop designs (i.e., designs 31171 and 31111, respectively.) As discussed above, both designs have exactly the same mean processing times in each machine. For example, the mean and variance of the processing time in machine #1 is 2.8492 minutes and 0.7591, respectively, for both low and highly loaded designs. The coefficient of variation of total processing times is 0.1170. Mean and variance of the lead time in each machine for low loaded design is approximately 4.1 minutes and 0.5, respectively. These statistics increase up to approximately 7.5 minutes and 0.7, respectively, for the highly loaded design. The 80% increase in the mean (from 12.4691 to 22.5189) of the total lead time is accompanied by a 545% increase in the variance (from 19.6715 to 126.7942).

Table 5.1 Comparison of the variability of low and highly loaded designs.

	Lead Time				**Processing Time**			
	M/C #1	**M/C #1**	**M/C #1**	**Total**	**M/C #1**	**M/C #1**	**M/C #1**	**Total**
	Design 31171 (low loaded system)							
Mean	4.1817	4.1419	4.1454	**12.4691**	2.8492	2.8430	2.8469	**8.5390**
Variance	5.3243	5.0087	4.9977	**19.6715**	0.7591	0.7462	0.7646	**2.2839**
CV	0.5518	0.5403	0.5393	**0.3557**	0.3058	0.3038	0.3072	**0.1170**
	Design 31111 (highly loaded system)							
Mean	7.6131	7.3943	7.5115	**22.5189**	2.8492	2.8430	2.8469	**8.5390**
Variance	31.3240	27.8359	28.2791	**126.7942**	0.7591	0.7462	0.7646	**2.2839**
CV	0.7352	0.7135	0.7080	**0.5000**	0.3058	0.3038	0.3072	**0.1170**

To summarize, increasing the load level of a system in job-shops increases the length of the transient period due to significant increase in the variability of the system.

5.5 The effect of reliability

We included only one level for the reliability of machines, and chose it as 90% availability. It is experimentally found that introducing *unreliable machines* to the system *increases the length of the transient period*. As discussed in Section 4.6, allowing unreliable machines introduces additional variability to the system. This variability is much more greater for job-shop systems than that of serial line systems. Consider, for example, design 31111 in Figure 5.1 (a) whose unreliable version (311111221) is given in Figure 5.3 (a). MSER-5 heuristic suggests a truncation point of 25 and 149 for these designs, respectively. Similar observation is made for other designs, as well (compare Figures 5.1 to 5.3, 5.2 to 5.4, and 5.7 to 5.8).

As was found for the serial line system, type of breakdown also has significant effect on the length of the transient period. *Frequent but short breakdowns attains shorter transient period than rare but long breakdowns*. The degree of variability introduced by rare but long breakdowns is significantly higher than that of frequent but short breakdowns. Design 311111221 in Figure 5.3 (a) reaches steady-state at the 149th observation whereas its counterpart, i.e., design 311111222 in Figure 5.5 (a) reaches steady-state at the 2217th observation. Similar observation is made for other designs, as well (compare Figures 5.3 to 5.5, 5.4 to 5.6, and 5.8 to 5.9).

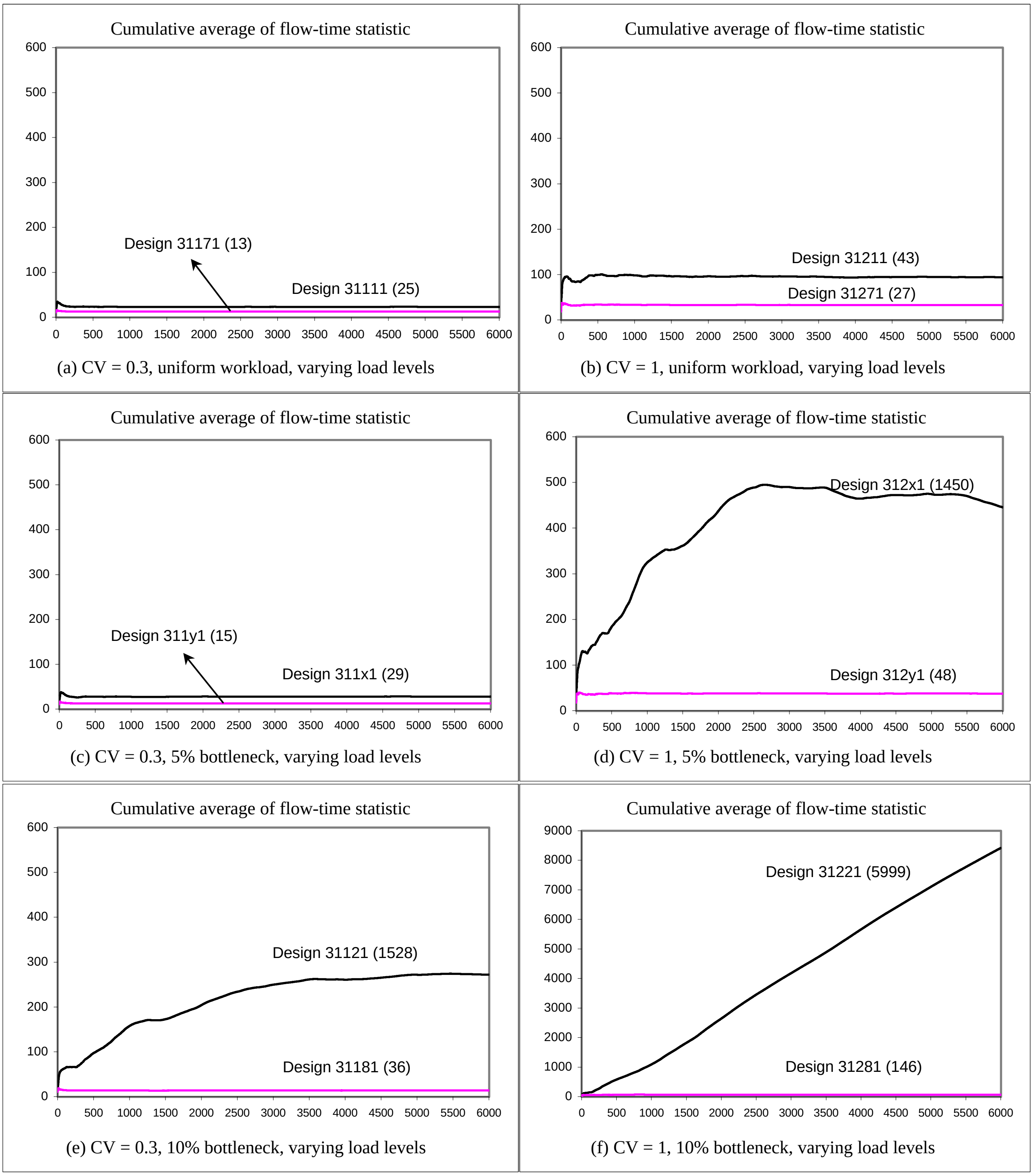
5.6 The effect finite buffer capacities

Finally, we introduced capacitated buffers in a job-shop system as in serial lines and investigated the effect of limiting the buffer capacities to 10. It is experimentally found that systems with finite buffer capacities attains *longer transient period* than that of systems with no buffer phenomenon. Figure 5.10 presents the capacitated (in the buffer capacity sense) versions of the designs in Figure 5.1. Apparently capacitated designs have longer transient period than their uncapacitated counterparts.

Notice that, it was assumed for job-shop designs that there is no buffer in the system. Hence, there was only one type of coupling event, which is in the form of starvation. However, when we introduce a capacitated buffer to the system we are in effect allowing machine blockages, as well. Therefore, the degree of coupling for capacitated systems is higher, which in turn will result with longer transient period.

This result seems to be different from that of serial lines. However, when considered carefully, the results are consistent. In serial lines, there assumed to be only finite capacitated intermediate storage buffers. However, in job-shops we assume that there exists an infinite storage area plus finite capacitated buffers in front of each machine. Adding finite buffers also adds a second dimension of coupling, i.e., blocking. This increase in coupling causes an increase in transient period, as well.

For instance, consider design `31111` in Figure 5.10 (a). The capacitated version of this design is also given in Figure 5.10 (c). The uncapacitated and capacitated designs reaches steady-state at 25^{th} and 134^{th} observation, respectively. The mean and variance of the uncapacitated system is found as 22.514 and 126.79, respectively. And, for the capacitated system they are found as 38.283 and 193.14, respectively.



(a) CV = 0.3, uniform workload, varying load levels

(b) CV = 1, uniform workload, varying load levels

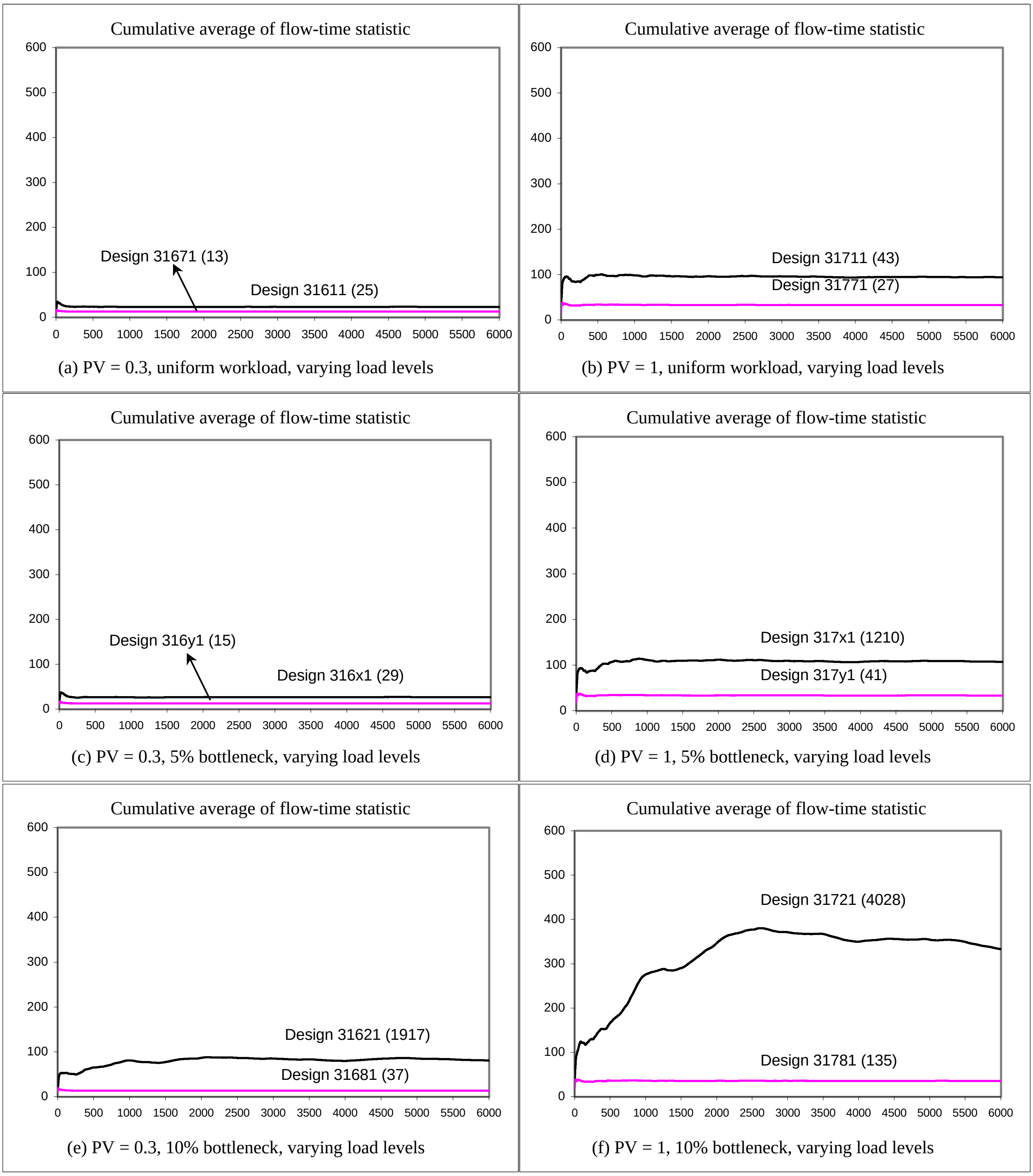
(c) CV = 0.3, 5% bottleneck, varying load levels

(d) CV = 1, 5% bottleneck, varying load levels

(e) CV = 0.3, 10% bottleneck, varying load levels

(f) CV = 1, 10% bottleneck, varying load levels

Figure 5.1 Experimental results for 3-machine job-shop containing all reliable machines with a total processing time of 3 minutes per job.



(a) PV = 0.3, uniform workload, varying load levels

(b) PV = 1, uniform workload, varying load levels

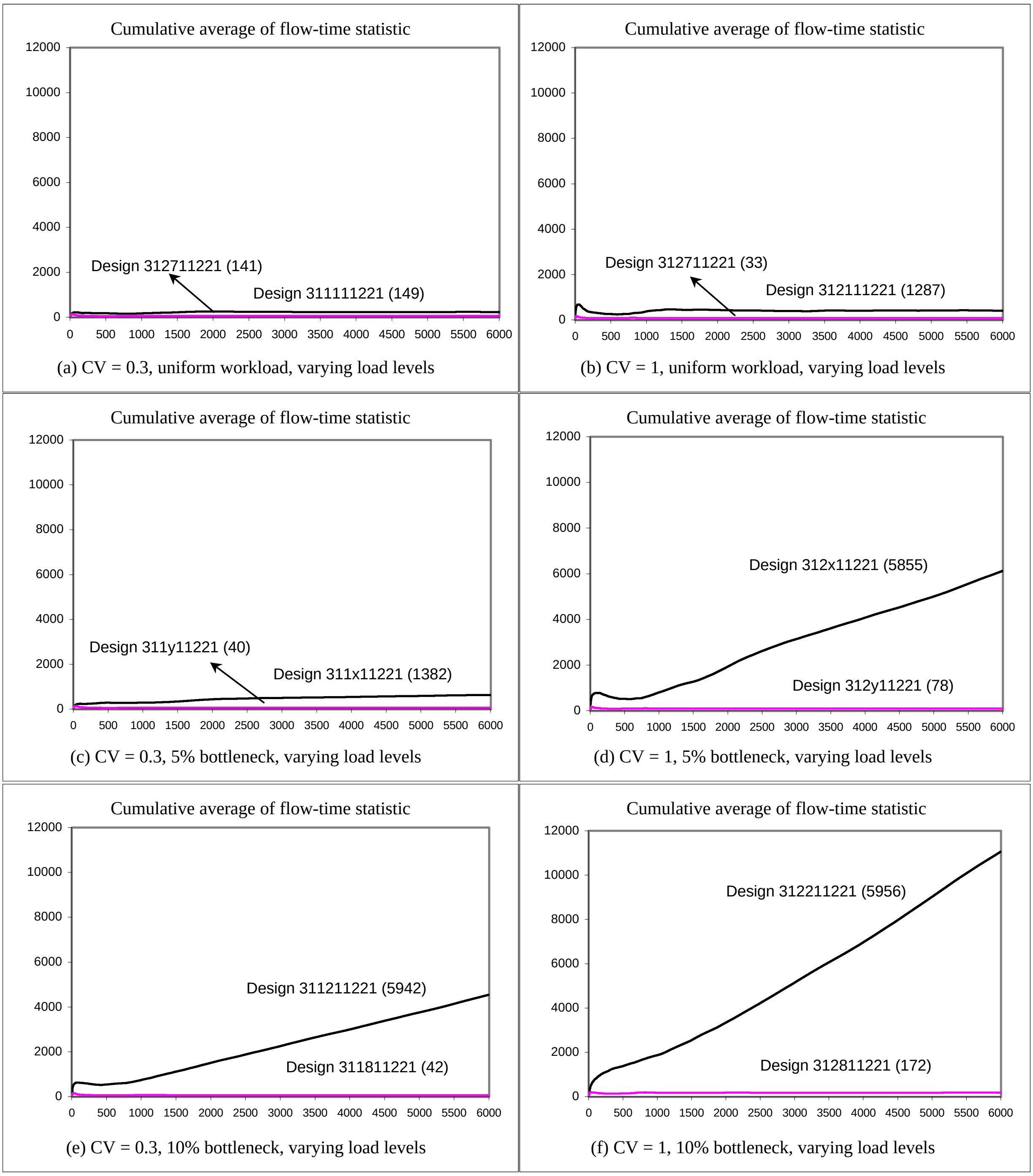
(c) PV = 0.3, 5% bottleneck, varying load levels

(d) PV = 1, 5% bottleneck, varying load levels

(e) PV = 0.3, 10% bottleneck, varying load levels

(f) PV = 1, 10% bottleneck, varying load levels

Figure 5.2 Experimental results for 3-machine job-shop containing all reliable machines with a total processing time of 3 minutes per job.



(a) CV = 0.3, uniform workload, varying load levels

(b) CV = 1, uniform workload, varying load levels

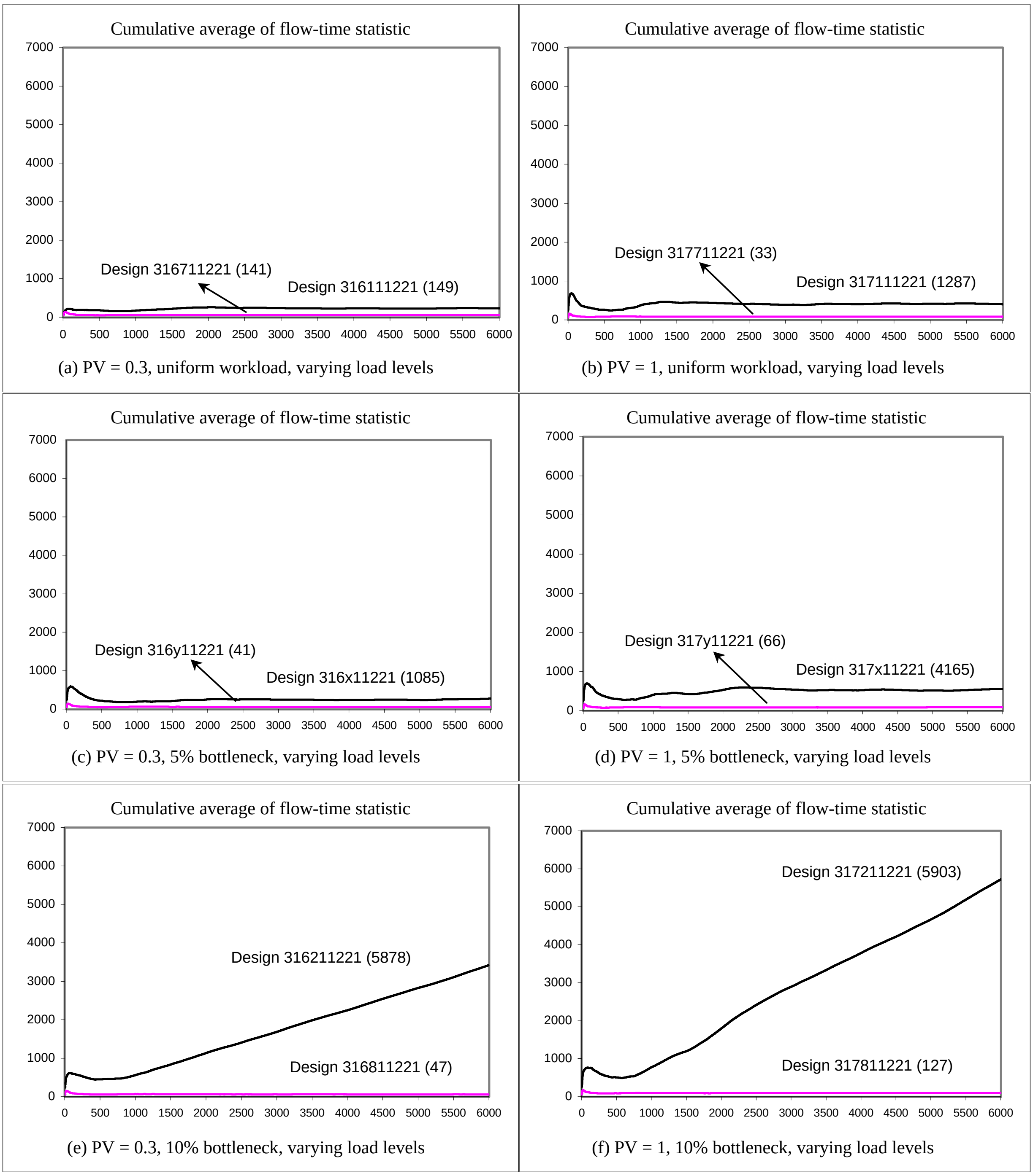
(c) CV = 0.3, 5% bottleneck, varying load levels

(d) CV = 1, 5% bottleneck, varying load levels

(e) CV = 0.3, 10% bottleneck, varying load levels

(f) CV = 1, 10% bottleneck, varying load levels

Figure 5.3 Experimental results for 3-machine job-shop containing 90% unreliable machines with frequent breakdown/short repair times and a total processing time of 3 minutes per job.



(a) PV = 0.3, uniform workload, varying load levels

(b) PV = 1, uniform workload, varying load levels

(c) PV = 0.3, 5% bottleneck, varying load levels

(d) PV = 1, 5% bottleneck, varying load levels

(e) PV = 0.3, 10% bottleneck, varying load levels

(f) PV = 1, 10% bottleneck, varying load levels

Figure 5.4 Experimental results for 3-machine job-shop containing 90% unreliable machines with frequent breakdown/short repair times and a total processing time of 3 minutes per job.

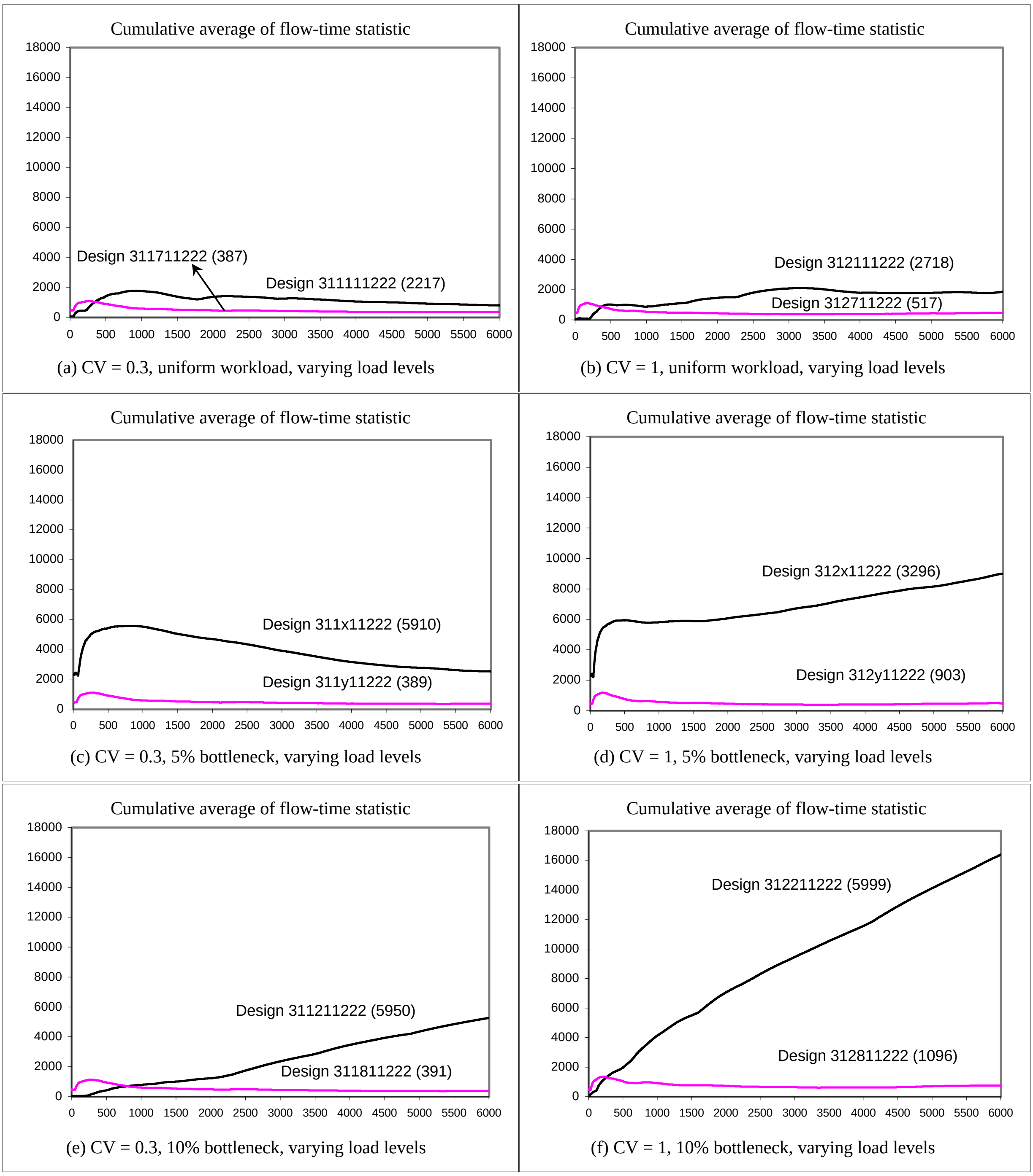
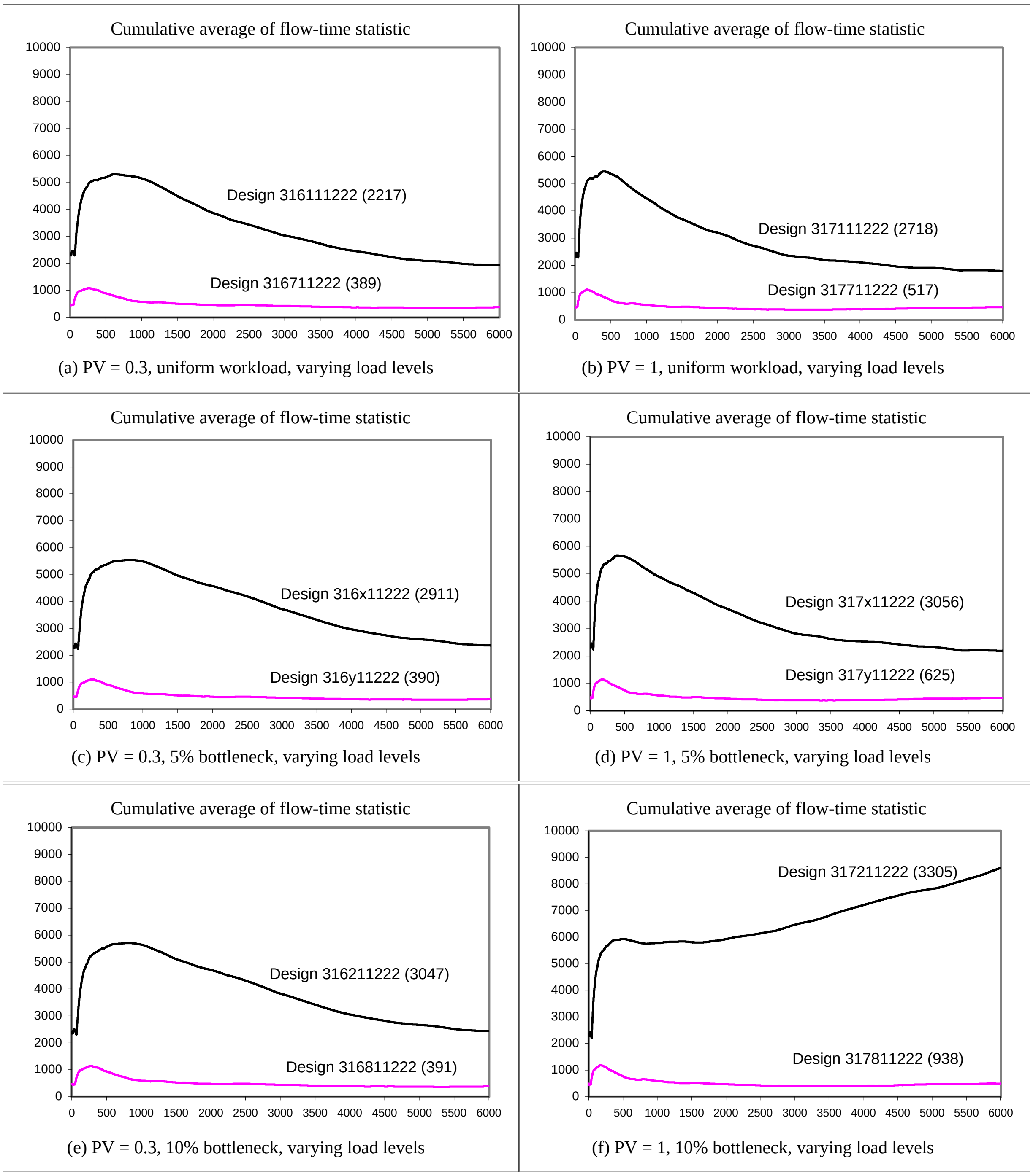


Figure 5.5 Experimental results for 3-machine job-shop containing 90% unreliable machines with rare breakdown/long repair times and a total processing time of 3 minutes per job.



(a) PV = 0.3, uniform workload, varying load levels

(b) PV = 1, uniform workload, varying load levels

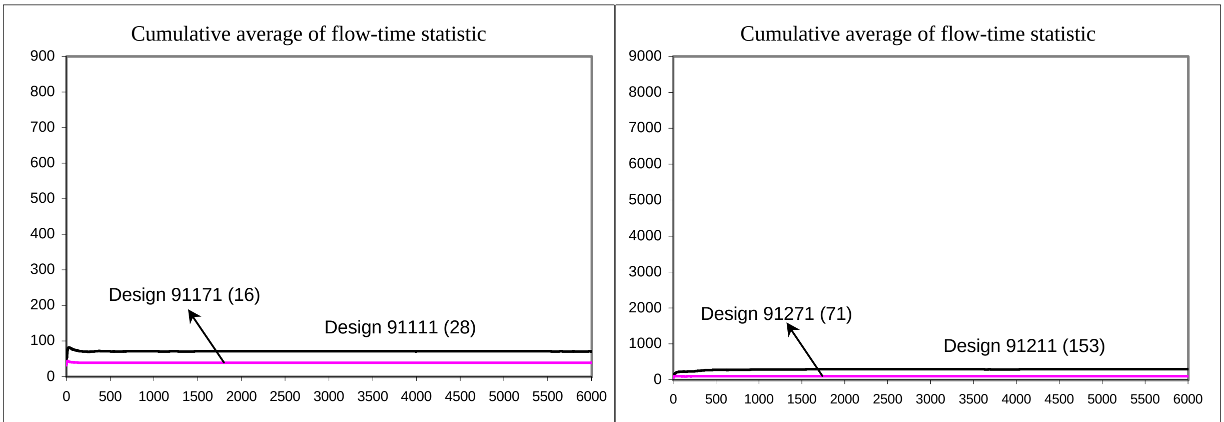
(c) PV = 0.3, 5% bottleneck, varying load levels

(d) PV = 1, 5% bottleneck, varying load levels

(e) PV = 0.3, 10% bottleneck, varying load levels

(f) PV = 1, 10% bottleneck, varying load levels

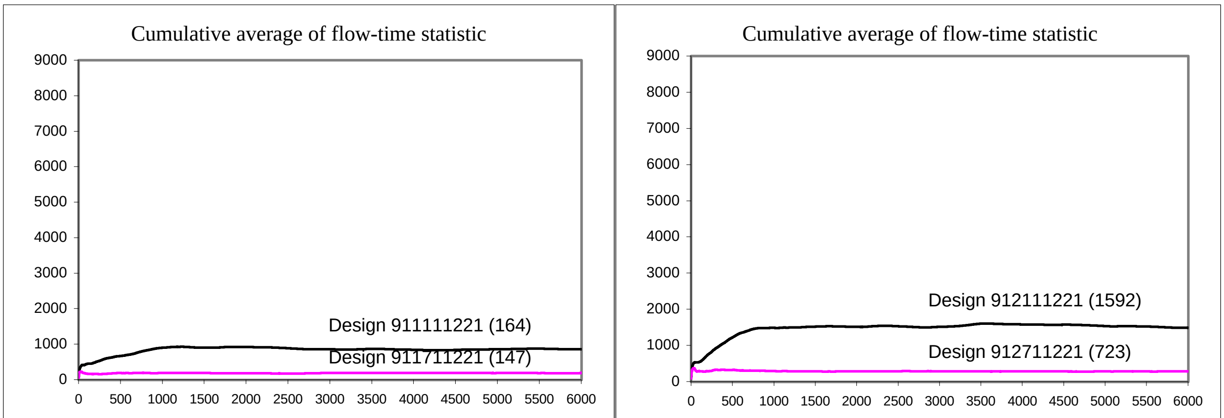
Figure 5.6 Experimental results for 3-machine job-shop containing 90% unreliable machines with rare breakdown/long repair times and a total processing time of 3 minutes per job.



(a) CV = 0.3, uniform workload, varying load levels

(b) CV = 1, uniform workload, varying load levels

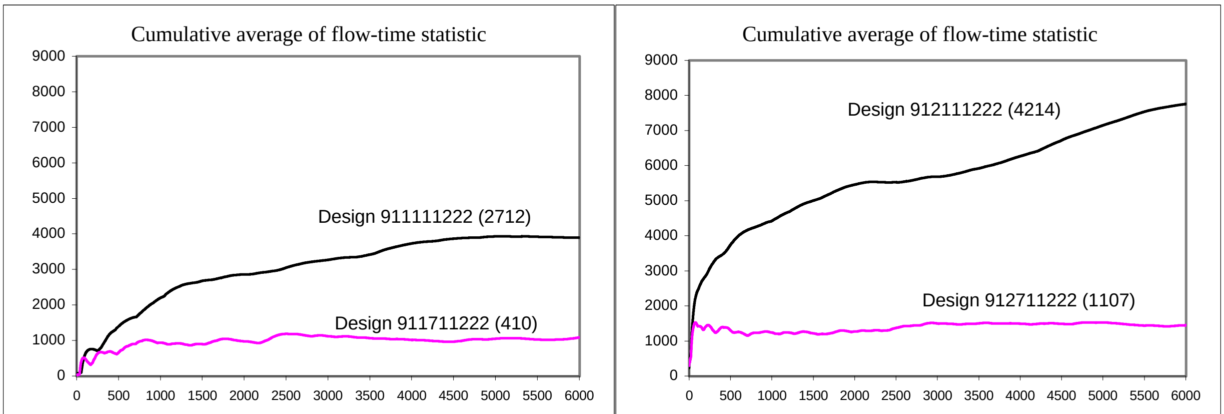
Figure 5.7 Experimental results for 9-machine job-shop containing all reliable machines with a total processing time of 9 minutes per job.



(c) CV = 0.3, uniform workload, varying load levels

(d) CV = 1, uniform workload, varying load levels

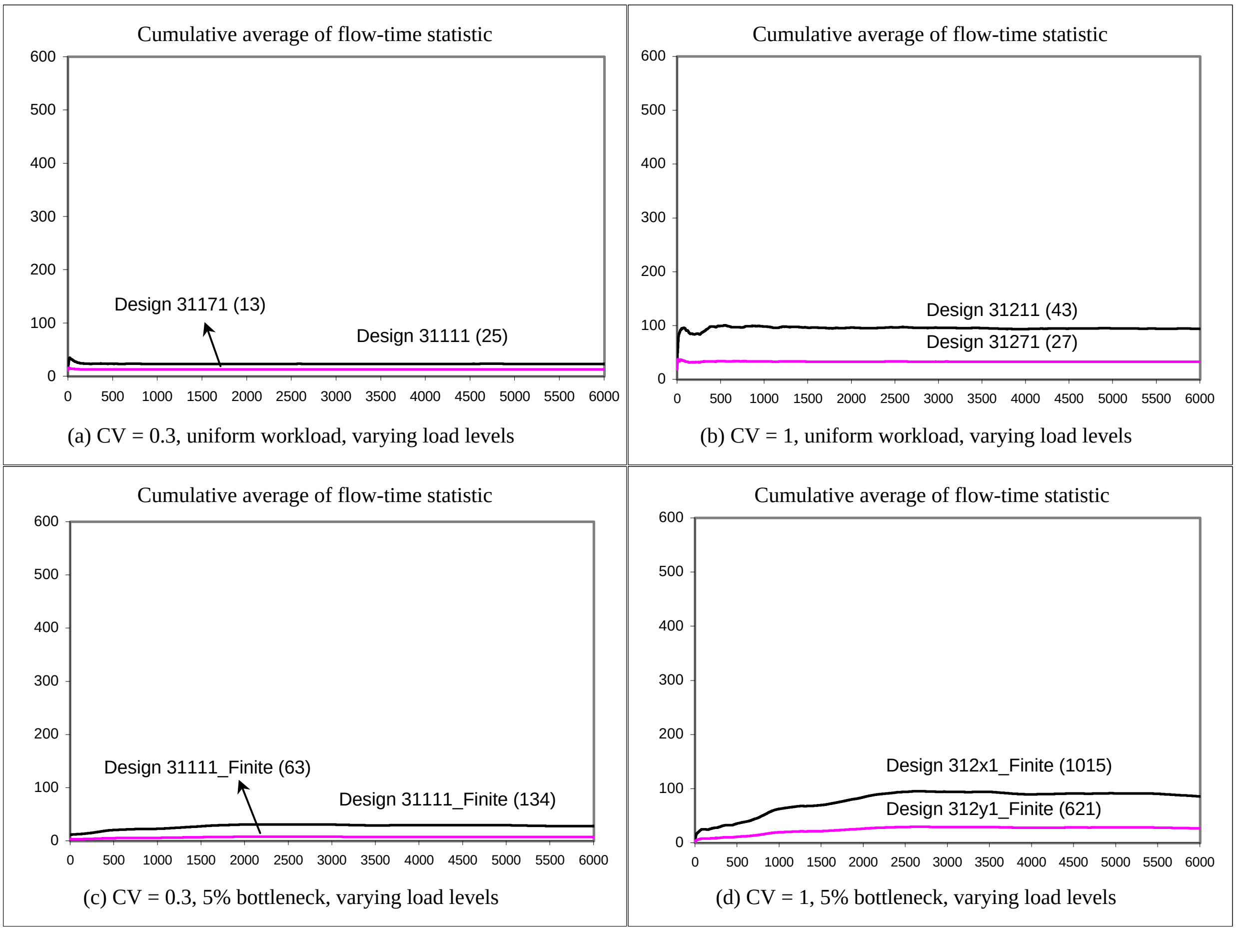
Figure 5.8 Experimental results for 9-machine job-shop containing 90% unreliable machines with frequent breakdown/short repair times and a total processing time of 9 minutes per job.



(e) CV = 0.3, uniform workload, varying load levels

(f) CV = 1, uniform workload, varying load levels

Figure 5.9 Experimental results for 9-machine job-shop containing 90% unreliable machines with rare breakdown/long repair times and a total processing time of 9 minutes per job.



(a) CV = 0.3, uniform workload, varying load levels

(b) CV = 1, uniform workload, varying load levels

(c) CV = 0.3, 5% bottleneck, varying load levels

(d) CV = 1, 5% bottleneck, varying load levels

Figure 5.10 Experimental results for 3-machine job-shop containing all reliable machines with a total processing time of 3 minutes per job (Finite vs. infinite buffers).

6 CONCLUSIONS

Simulation studies are often conducted for stochastic systems that are hard to analyze analytically. Since random number sequences are used to represent the stochastic nature of the process, simulation studies do not give exact solutions for the systems under consideration. Simulation results are particular realizations (or estimates) of some performance measure(s) of interest. Careful statistical analyses of the simulation outputs should be made to have better (true) estimates.

One of the important problems in the output analysis literature of nonterminating simulations is the initial transient problem, which is a result of initializing the simulation run in a state that is unrepresentative of the steady-state conditions. Although there is much literature about this problem, we did not encounter any research explicitly studying the *behavior* of the initial transient period. There are studies that investigate the length of the transient period when different initial conditions are chosen (see, for example, Fishman, 1972; Kelton, 1989; Kelton and Law, 1985; Madansky, 1976; Murray and Kelton, 1988b). However, these do not totally match with our study. In this thesis, we studied the behavior of initial transient period for nonterminating manufacturing simulations, particularly, serial production lines and job-shop production systems. The results for serial lines and job-shops are summarized in Tables 6.1 and 6.2, respectively.

Table 6.1 Summary of the results for serial production lines.

Factors				**Transient Period**
Variability of processing time			Increase	Increase
Line length			Increase	Increase
Bottleneck		Low CV	Increase	Decrease (slight)
		High CV	Increase	Increase
		Low PV	Increase	Decrease (slight)
		High PV	Increase	No change
Load level		Low CV	Increase	Increase (slight)
		High CV	Increase	Decrease
		Low PV	Increase	Increase (slight)
		High PV	Increase	No change
Reliability	Existence of unreliable machines	Low CV, PV	-	Varies according to type of breakdowns
		High CV, PV	-	No change
	Depth of breakdown	Low CV, PV	Increase	Increase (slight)
		High CV, PV	Increase	No change
	Type of breakdown	Low CV, PV	-	$T_{p\,(RARE)} > T_{p\,(FREQUENT)}$
		High CV, PV	-	No change
Buffer capacity			Increase	Increase

Table 6.2 Summary of the results for job-shop production systems.

Factors				**Transient Period**
Variability of processing time			Increase	Increase
System size			Increase	Increase
Bottleneck		Low CV	Increase	Increase
		High CV	Increase	Increase
		Low PV	Increase	Increase
		High PV	Increase	Increase
Load level		Low CV	Increase	Increase
		High CV	Increase	Increase
		Low PV	Increase	Increase
		High PV	Increase	Increase
Reliability	Existence of unreliable machines	Low CV, PV	-	Increase
		High CV, PV	-	Increase
	Type of breakdown	Low CV, PV	-	$T_{p\,(RARE)} > T_{p\,(FREQUENT)}$
		High CV, PV	-	$T_{p\,(RARE)} > T_{p\,(FREQUENT)}$
Capacitated buffer			-	$T_{p\,(CAP.)} > T_{p\,(UNCAP.)}$

Before continuing with a discussion of the results, the following should be noted. As Conway (1963) suggested most people uses the same transient period in comparing alternative system designs. Actually, this is the true behavior from statistical comparison point of view. However, from efficiency point of view, this might cause two problems.

An underestimation will occur if a decision about the length of the transient period is based on a system that has a relatively short transient period (best case scenario). This might remain significant amount of initialization bias in other designs that have significantly longer transient periods. However, on the other hand, if this decision is based on a system that has a relatively very long transient period (worst case scenario), then overestimation might occur, which would result with unnecessary loss of data from other designs. Therefore, from practical point of views, there should exist a tradeoff between statistical reliability and practical efficiency. Practitioners should take care of this tradeoff and should not base their decisions about the length of transient period on a single pilot run. Rather, they should try much more designs and have an idea of the tradeoff and than conclude about the length of the transient period.

Considering both serial line and job-shop results we can make the following discussion:

1. As the variability of processing times is increased the length of the transient period increases significantly both for serial line and job-shop production systems. Additionally, variability is found to be the most significant factor among all factors affecting the transient period. If a system that has a highly variable processing times (i.e., $CV \geq 1$) were to be analyzed, then the analyst should take a fairly long run to have enough observations from the steady-state distribution. The above discussion about deciding the length of transient period based on best and worst case scenarios is apparent in the variability case. Based on our experimental results, the increase in the length of the transient period ranges from approximately 3000 times to approximately 1.3 times as the variability of processing times is increased from 0.3 to 2.5. Therefore, the analyst should decide on the tradeoff instead of a single best or worst case.
2. Increasing the system size for both serial line and job-shop production systems also increases the length of the transient period. System size is increased (decreased) by adding (removing) stages in a serial line and by adding (removing) machines in a job-shop. Therefore, an analyst studying a system that contains several entities should expect a long transient period.

3. The existence of bottleneck machines given a constant workload has a more complicated effect on the length of the transient period. For job-shop systems, forming bottleneck machines and further increasing the depth of the bottleneck simply increases the length of the transient period (maximum increase is approximately 76 times that of uniform system, i.e., compare designs 31611 and 31621 in Figure 5.2). However, for serial line systems, introducing bottleneck increases the length of transient period only in the high CV case, where there occurs no change in the high PV case. In the low variability case (either in the form of CV or PV), the length of the transient period slightly decreases by increasing the depth of bottleneck. In practice, if uniform systems were to be compared to systems containing bottleneck in a constant workload case and if the overall variability of the system can be assumed to be low, then the transient period for each system can be assumed same, although a slight decrease is observed in our results. However, if the overall variability of the system is high, then long transient periods would be expected.
4. System load level, which is measured by the mean processing time per job in serial lines and by the arrival rate in job-shops, also has complicated effect on the length of the transient period. For job-shop systems, increasing the load of the system also increases the length of the transient period (maximum increase is approximately 141 times that of uniform system, i.e., compare designs 311211221 and 311811221 in Figure 5.2). For serial line systems, increases the length of transient period only in low variability case (either measured by CV or PV). However, the behavior changes for the high CV and PV cases. Transient period decreases in the high CV case, whereas there occurs no change in the high PV case.
5. The existence of unreliable machines in job-shops, which is a more realistic case than all reliable machines, increases the length of transient period. However, for serial lines, the length of transient period remains same as the reliable versions in the high variability case (either measured by CV or PV). Type of breakdown affects the effect of unreliability. For instance, for 90% availability, if frequent breakdowns/ short repair times is allowed than then there occurs a slight decrease in the transient period. However, if, for exactly the same system, rare breakdowns/long repair times are allowed than there occurs an increase in the transient period. Increasing the depth of

unreliability increases the length of transient period in the low variability case, whereas it has no effect in the high variability case. Additionally, breakdown type has no effect on the length of transient period in high variability case, whereas rare but long breakdowns attains a longer transient period than frequent but short breakdowns. Based on our experiments, the increase in the length of the transient period ranges from 2 times to 10 times as we increase the size of the system from 3 to 9 machines.

6. For serial line systems, increasing the buffer capacities also increases the length of transient period (e.g., it can be as much as 750 times, compare designs `31111` and `31114` in Figure 4.1). Similar conclusion can be made for job-shops: introducing a finite capacity buffer increases the transient period.

A very general recommendation about comparing the transient periods of two or more alternative system designs can be given as follows: The system having more variable output sequences will apparently have longer transient periods. Then one should first investigate the change in the variability of the output sequences. If any of the factors are suspected to introduce additional variability to a system, then a longer transient period should be expected for that design. For example, including unreliable machines to a system will introduce additional variability to the overall process. However, the degree of this additional variability is determined by the depth and type of unreliability. Hence, in comparing alternative system designs one should truncate exactly the same amount of data from each design, which should be determined from the most variable design.

If the variability of alternative designs are very close to each other but one of them has much more entities (i.e., machines, complicated material handling, etc.) then the analyst should base his decision about the length of the transient period on this particular design. The degree of coupling in manufacturing systems is an important factor that affects the transient period.

For extreme cases, intelligent initialization techniques might help reducing the length of the transient period, which, if works, apparently will save computer run time.

It is also found that, in most of the cases, both cumulative averages plots and MSER heuristic results comply with each other, with cumulative averages usually overestimating the transient period. Since MSER heuristic is an objective criterion and is very simple and computationally efficient, we recommend using this heuristic for determining the length of transient period. However, care must be given to the remove any outliers from the sequence, which otherwise would lead the analyst to wrong conclusions. Moreover, if there is enough time to use both techniques simultaneously, then it would be preferable.

For future research, we identify the following topics:

- Other systems such as distribution systems, inventory, network, and military applications can be considered within the provided framework.
- As an extension of our study and studies that might be done as stated in the above item, a research can be conducted to develop a formula for determining the length of the transient period which is a function of system parameters.

REFERENCES

1. Adlakha, V. G. and Fishman, G. S. (1982). "Starting and stopping rules for simulations using a priori information," *European Journal of Operational Research*, **10 (4)**, 379-394.

2. Altıok, T. (1997). *Performance Analysis of Manufacturing Systems*, Springer-Verlag.

3. Anderson, T. W. (1971). *The Statistical Analysis of Time-Series*, Wiley.

4. Asmussen, S., Glynn, P. W., and Thorisson, H. (1992). "Stationarity detection in the initial transient problem," *ACM Transactions on Modeling and Computer Simulation*, **2**, 130-157.

5. *AutoMod Technical Support*, Personal communication, July 2001.

6. *AutoMod User's Manual*, Vol. 1, AutoSimulations, Inc., June 1999.

7. Banks, J., Carson II, J. S., and Nelson B. L. (1996). *Discrete-Event System Simulation*, Prentice Hall.

8. Blomqvist, N. (1970). "On the transient behavior of *GI*/*G*/1 waiting times," *Skand. Aktuarietidskr.*, 118-129.

9. Buzacott, J. A. and Hanifin, L. E. (1978). "Models of automatic transfer lines with inventory banks – a review and comparison," *IIE Transactions*, **10**, 197-207.

10. Buzacott, J. A., and Shanthikumar, J. G. (1992). "Models of production systems," in *Handbook of Industrial Engineering*, G. Salvendy (ed.), 2nd ed., Wiley.

11. Buzacott, J. A., and Shanthikumar, J. G. (1993). *Stochastic Models of Manufacturing Systems*, Prentice Hall.

12. Cash, C. R., Nelson, B. L., Long, J. M., Dippold, D. G., and Pollard, W. P. (1992). "Evaluation of tests for initial-condition bias," in *Proceedings of the 1992 Winter*

Simulation Conference, J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson (eds.), 577-585.

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

14. Chow, W. M. (1990). *Assembly Line Design: Methodology and Applications*, Marcel Dekker.

15. Conway, R. W. (1963). "Some tactical problems in digital simulation," *Management Science*, **10**, 47-61.

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

17. Crane, M. A. and Iglehart, D.L. (1974a). "Simulating stable stochastic systems, I: General multi-server queues," *Journal of ACM*, **21**, 103-113.

18. Crane, M. A. and Iglehart, D.L. (1974b). "Simulating stable stochastic systems, II: Markov chains," *Journal of ACM*, **21**, 114-123.

19. Crane, M. A. and Iglehart, D.L. (1975). "Simulating stable stochastic systems, III: Regenerative processes and discrete-event simulation," *Operations Research,* **23**, 33-45.

20. Crane, M. A. and Lemoine, A. J. (1977). *An Introduction to the Regenerative Method for Simulation Analysis*, Vol 4., Springer-Verlag.

21. D'Angelo, A., Gastaldi, M., and Levialdi, N. (2000). "Production variability and shop configuration: An experimental analysis," *International Journal of Production Economics*, **68**, 217-228.

22. Dallery, Y. and Gershwin, S. B. (1992). "Manufacturing flow line systems: A review of models and analytical results," *Queuing Systems*, **12**, 3-94.

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

24. Enns, S. T. (2000). "Evaluating shop-floor input control using rapid modeling," *International Journal of Production Economics*, **63 (3)**, 229-241.

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

26. Fishman, G. S. (1971). "Estimating sample size in computer simulation experiments," *Management Science*, **17**, 21-38.

27. Fishman, G. S. (1972). "Bias considerations in simulation experiments," *Operations Research*, **20**, 785-790.

28. Fishman, G. S. (1973a). "Statistical analysis for queuing simulations," *Operations Research*, **20 (3)**, 363-369.

29. Fishman, G. S. (1973b). *Concepts and Methods in Discrete-Event Simulation*, Wiley.

30. Fishman, G. S. (1978). *Principles of Discrete Event Simulation*, Wiley.

31. Gafarian, A. V., Ancker Jr., C. J., and Morisaku, T. (?). "The problem of initial transient in digital computer simulation," *Winter Simulation Conference Proceedings*, 49-51.

32. Gafarian, A. V., Ancker Jr., C. J., and Morisaku, T. (1978). "Evaluation of commonly used rules for detecting steady-state in computer simulation," *Naval Research Logistics Quarterly*, **25**, 511-529.

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

34. Glynn, P. W. and Heidelberger, P. (1991). "Analysis of initial transient deletion for replicated steady-state simulations," *Operations Research*, **10**, 437-443.

35. Glynn, P. W. and Heidelberger, P. (1992a). "Analysis of initial transient deletion for parallel steady-state simulations," *SIAM Journal on Scientific and Statistical Computing*, **13 (4)**, 904-922.

36. Glynn, P. W. and Heidelberger, P. (1992b). "Experiments with initial transient deletion for parallel, replicated steady-state simulations," *Management Science*, **38 (3)**, 400-418.

37. Goldsman, D., Schruben, L. W., and Swain, J. J. (1994). "Tests for transient means in simulated time series," *Naval Research Logistics*, **41**, 171-187.

38. Heidelberger, P. and Welch, P. D. (1983). "Simulation run length control in the presence of an initial transient," *Operations Research*, **31**, 1109-1114.

39. Hopp, W. J., and Spearman, M. L. (2000). *Factory Physics*, McGraw-Hill.

40. Jaynes, E. T. (1957). "Information theory and statistical mechanics," *Physical Review*, **106**, 620-630.

41. Kadıpaşaoğlu, S. N., Xiang, W., Hurley, S. F., and Khumawala, B. M. (2000). "A study on the effect of the extent and location of protective capacity in flow systems," *International Journal of Production Economics*, **63**, 217-228.

42. Kelton, W. D. (1980). "The startup problem in discrete-event simulation," Technical Report No. 80-1, Department of Industrial Engineering, University of Wisconsin.

43. Kelton, W. D. (1985). "Transient exponential-Erlang queues and steady-state simulation," *Communications of ACM*, **28**, 741-749.

44. Kelton, W. D. (1986). "Replication splitting and variance for simulating discrete-parameter stochastic process," *Operations Research Letters*, **4**, 275-279.

45. Kelton, W. D. (1989). "Random initialization methods in simulation," *IIE Transactions*, **21 (4)**, 355-367.

46. Kelton, W. D. and Law, A. M. (1981). "An analytical evaluation of alternative strategies in steady-state simulation," Technical Report 81-3, Department of Management Information Systems, University of Arizona.

47. Kelton, W. D., and Law, A. M. (1983). "A new approach for dealing with the startup problem in discrete event simulation," *Naval Research Logistics Quarterly*, **30**, 641-658.

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

49. Kleijnen, J. P. C. (1975). *Statistical Tools for Simulation Practitioners*, Dekker.

50. Kleijnen, J. P. C. (1984). "Statistical analysis of steady-state simulations: survey of recent progress," *European Journal of Operational Research*, **17**, 150-162.

51. Knott, K., and Sury, R. J. (1987). "A study on work-time distributions on unpaced tasks," *IIE Transactions*, **19**, 50-55.

52. Law, A.M. (1975). "A comparison of two techniques for determining the accuracy of simulation output," Technical Report No. 75-11, Department of Industrial Engineering, University of Wisconsin.

53. Law, A.M. (1977). "Confidence intervals in discrete event simulation: A comparison of replication and batch means," *Naval Research Logistics Quarterly*, **24**, 667-678.

54. Law, A. M. (1980). "Statistical analysis of the output data from terminating simulations," *Naval Research Logistics Quarterly*, **27**, 131-143.

55. Law, A. M. (1984). “Statistical analysis of the simulation output data,” *Operations Research*, **31**, 983-1029.

56. Law, A. M. and Carson, J. S. (1979). “A sequential procedure for determining the length of a steady-state simulation,” *Operations Research*, **27**, 1011-1025.

57. Law, A. M. and Kelton, W. D. (1982). “Confidence intervals for steady-state simulations, II: A survey of sequential procedures,” *Management Science*, **28**, 550-562.

58. Law, A. M. and Kelton, W. D. (1984). “Confidence intervals for steady-state simulations, I: A survey of fixed sample size procedures,” *Operations Research*, **32**, 1221-1239.

59. Law, A. M., and Kelton, W. D. (2000). *Simulation Modeling and Analysis*, 3rd ed., McGraw Hill.

60. Lehmer, D. H. (1951). “Mathematical methods in large-scale computing units,” *Ann. Comp. Lab.*, Harvard University, **26**, 141-146.

61. Ma, X. and Kochhar, A. K. (1993). “A comparison study of two tests for detecting initialization bias in simulation output,” *Simulation*, 94-101.

62. Madansky, A. (1976). “Optimal initial conditions for a simulation problem,” *Operations Research*, **24 (3)**, 572-577.

63. Montgomery, D. C. (1984). *Design and Analysis of Experiments*, 2nd ed., Wiley.

64. Moran, P. A. P. (1959). *The Theory of Storage*, Wiley.

65. Morisaku, T. (1976). “Techniques for data truncation in digital computer simulation,” Ph.D. dissertation, Department of Industrial and Systems Engineering, University of Southern California.

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

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

68. Murray, J. R. and Kelton, W. D. (1988b), “Initializing for bias reduction: Some analytical results,” in *Proceedings of the 1988 Winter Simulation Conference*, M. Abrams, P. Haigh, and J. Comfort (eds.), 546-548.

69. Nelson, B. L. (1990). “Variance reduction in the presence of initial-bias,” *IIE Transactions*, **22**, 340-350.

70. Nelson, B. L. (1992). “Initial-condition bias,” in *Handbook of Industrial Engineering*, G. Salvendy (ed.), 2nd ed., Wiley.

71. Papadopoulos, H. T., and Heavey, C. (1996). “Queuing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines,” *European Journal of Operations Research*, **92**, 1-27.

72. Sabuncuoğlu, I., and Karapınar, H. Y. (1999). “Analysis of order review/release problems in production systems,” *International Journal of Production Economics*, **62**, 259-279.

73. Schruben, L. W. (1981). “Control of initialization bias in multivariate simulation response,” *Communications of the ACM*, **24**, 246-252.

74. Schruben, L. W. (1982). “Detecting initialization bias in simulation output,” *Operations Research*, **30**, 569-590.

75. Schruben, L. W. and Goldsman, D. (1985). “Initialization effects in computer simulation experiments,” Technical Report No. 594, School of Operations Research and Industrial Engineering, Cornell University.

76. Schruben, L. W., Singh, H., and Tierney, L., (1983). “Optimal test for initialization bias in simulation output,” *Operations Research*, **31**, 1167-1178.

77. Snell, M. and Schruben, L. W. (1985). “Weighting simulation data to reduce initialization effects,” *IIE Transactions*, **17**, 354-365.

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

79. Vassilacopoulus, G. (1989). “Testing for initialization bias in simulation output,” *Simulation*, **52**, 151-153.

80. Welch, P. D. (1981). “On the problem of the initial transient in steady-state simulation,” IBM Watson Research Center, Yorktown Heights.

81. Welch, P. D. (1982). “A graphical approach to the initial transient problem in steady-state simulations,” in *Proceedings of the 10th IMACS World Congress on Systems, Simulation, and Scientific Computation*, Montreal, 219-221.

82. Welch, P. D. (1983). “The statistical analysis of simulation results,” in *The Computer Performance Modeling Handbook*, S. S. Lavenberg (Ed.), Academic Press, New York, 268-328.

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

84. White Jr, K. P. (2001). Personal communication.

85. White Jr., K. P., Cobb, M. J., and Spratt, S. C. (2000), “A comparison of five steady-state truncation heuristics for simulation,” in *Proceedings of the 2000 Winter*

Simulation Conference, J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick (eds.), 755-760.

86. Whitt, W. (1991). "The efficiency of one long run versus independent replications in steady-state simulation," *Management Science*, **37 (6)**, 645-666.

87. Wilson, J. R., and Pritsker, A. A. B. (1978a). "A survey of research on the simulation startup problem," *Simulation*, **31**, 55-58.

88. Wilson, J. R., and Pritsker, A. A. B. (1978b), "Evaluation of startup policies in simulation experiments," *Simulation*, **31**, 79-89.

APPENDIX A. RANDOM NUMBER STREAMS

Table A.1 Usage purpose of different random number streams and their reference points in serial production line experiments.

Stream Number	Skip First	Usage Purpose
Streams 1, 11, 21, 31, 41	1.000.000, 11.000.000, 21.000.000, 31.000.000, 41.000.000	Operation times on 1st machine for run 1, 2, 3, 4, 5.
" 2, 12, 22, 32, 42	2.000.000, 12.000.000, 22.000.000, 32.000.000, 42.000.000	" " " 2nd " " " 1, 2, 3, 4, 5.
.	.	.
.	.	.
.	.	.
" 10, 20, 30, 40, 50	10.000.000, 20.000.000, 30.000.000, 40.000.000, 50.000.000	" " " 10th " " " 1, 2, 3, 4, 5.
Streams 51, 61, 71, 81, 91	51.000.000, 61.000.000, 71.000.000, 81.000.000, 91.000.000	Mean breakdown time of 1st machine for run 1, 2, 3, 4, 5.
" 52, 62, 72, 82, 92	52.000.000, 62.000.000, 72.000.000, 82.000.000, 92.000.000	" " " " 2nd " " " 1, 2, 3, 4, 5.
.	.	.
.	.	.
.	.	.
" 60, 70, 80, 90, 100	60.000.000, 70.000.000, 80.000.000, 90.000.000, 100.000.000	" " " " 10th " " " 1, 2, 3, 4, 5.
Streams 101, 111, 121, 131, 141	101.000.000, 111.000.000, 121.000.000, 131.000.000, 141.000.000	Mean repair time of 1st machine for run 1, 2, 3, 4, 5.
" 102, 112, 122, 132, 142	102.000.000, 112.000.000, 122.000.000, 132.000.000, 142.000.000	" " " " 2nd " " " 1, 2, 3, 4, 5.
.	.	.
.	.	.
.	.	.
" 110, 120, 130, 140, 150	110.000.000, 120.000.000, 130.000.000, 140.000.000, 150.000.000	" " " " 10th " " " 1, 2, 3, 4, 5.

Table A.2 Usage purpose of different random number streams and their reference points in job-shop experiments.

Stream Number	**Skip First**	**Usage Purpose**
Streams 1, 11, 21, 31, 41	1.000.000, 11.000.000, 21.000.000, 31.000.000, 41.000.000	Operation times on 1^{st} machine for run 1, 2, 3, 4, 5.
" 2, 12, 22, 32, 42	2.000.000, 12.000.000, 22.000.000, 32.000.000, 42.000.000	" " " 2^{nd} " " " 1, 2, 3, 4, 5.
.	.	.
.	.	.
.	.	.
" 10, 20, 30, 40, 50	10.000.000, 20.000.000, 30.000.000, 40.000.000, 50.000.000	" " " 10^{th} " " " 1, 2, 3, 4, 5.
Streams 51, 61, 71, 81, 91	51.000.000, 61.000.000, 71.000.000, 81.000.000, 91.000.000	Mean breakdown time of 1^{st} machine for run 1, 2, 3, 4, 5.
" 52, 62, 72, 82, 92	52.000.000, 62.000.000, 72.000.000, 82.000.000, 92.000.000	" " " " 2^{nd} " " " 1, 2, 3, 4, 5.
.	.	.
.	.	.
.	.	.
" 60, 70, 80, 90, 100	60.000.000, 70.000.000, 80.000.000, 90.000.000, 100.000.000	" " " " 10^{th} " " " 1, 2, 3, 4, 5.
Streams 101, 111, 121, 131, 141	101.000.000, 111.000.000, 121.000.000, 131.000.000, 141.000.000	Mean repair time of 1^{st} machine for run 1, 2, 3, 4, 5.
" 102, 112, 122, 132, 142	102.000.000, 112.000.000, 122.000.000, 132.000.000, 142.000.000	" " " " 2^{nd} " " " 1, 2, 3, 4, 5.
.	.	.
.	.	.
.	.	.
" 110, 120, 130, 140, 150	110.000.000, 120.000.000, 130.000.000, 140.000.000, 150.000.000	" " " " 10^{th} " " " 1, 2, 3, 4, 5.
Streams 151, 152, 153, 154, 155	151.000.000, 152.000.000, 153.000.000, 154.000.000, 155.000.000	Choosing the operation sequence of parts for run 1, 2, 3, 4, 5.
Streams 156, 157, 158, 159, 160	156.000.000, 157.000.000, 158.000.000, 159.000.000, 160.000.000	Interarrival time of parts to the system for run 1, 2, 3, 4, 5.

APPENDIX B. BREAKDOWN PARAMETERS

Parameters;

μ_U = mean uptime of the machine

μ_D = mean downtime of the machine

α_U = shape parameter of the uptime distribution

α_D = shape parameter of the downtime distribution

β_U = scale parameter of the uptime distribution

β_D = scale parameter of the downtime distribution

e = efficieny of the machine

Assumptions due to Law and Kelton 59:

- Uptime distribution is gamma
- Downtime distribution is gamma

Given:

- Shape parameter for uptime distribution is $\alpha_U = 0.7$
- Shape parameter for downtime distribution is $\alpha_D = 1.4$
- Mean downtime, μ_D
- Efficiency, e.

  Efficiency is defined as:

$$e = \frac{\mu_U}{\mu_U + \mu_D} \quad \text{(B.1)}$$

Then the scale parameters of uptime and downtime distributions can be obtained by using the following relation: "The mean of a gamma distribution is simply obtained by multiplying its shape and scale parameters."

For downtime distribution;

$$\mu_D = \alpha_D \cdot \beta_D \quad \text{(B.2)}$$

Rewriting equation (XXX.2) for β_D and substituting $\alpha_D = 1.4$ we obtain the scale parameter for downtime distribution as;

$$\beta_D = \frac{\mu_D}{1.4} \quad \text{(B.3)}$$

For uptime distribution;

$$\mu_U = \alpha_U \cdot \beta_U \qquad \text{(B.4)}$$

Solving equation (A.1) for μ_U we get,

$$\mu_U = \frac{e\mu_D}{1-e} \qquad \text{(B.5)}$$

Substituting μ_U and $\alpha_U = 0.7$ in (XXX.5) and rewriting (XXX.5) for β_U, we obtain the scale parameter for uptime distribution as;

$$\beta_U = \frac{e\mu_D}{0.7(1-e)} \qquad \text{(B.6)}$$

For instance, let e =90 % and μ_D = 1 hours. Then,

$$\beta_D = \frac{1}{1.4} = 0.714$$

$$\beta_U = \frac{(0.9)(1)}{0.7(1-0.9)} = 12.857$$

which states that the uptime distribution of machines should be modeled with gamma (0.7, 12.857) and that of downtime distribution with gamma (1.4, 0.714). Table B.1 lists these calculations for different efficiency (availability) rates and breakdown types.

Table B.1 Parameter selection for downtime and uptime distributions.

Availability	**MTBF**[a]	**MRT**[b]	**TST**[c]	**Breakdown Type**	**Parameters of gamma distribution**			
					α_U	β_U	α_D	β_D
90%	9	1	10	FBSR[d]	0.7	12.857	1.4	0.714
	90	10	100	RBLR[e]	0.7	128.571	1.4	7.143
80%	8	2	10	FBSR	0.7	11.429	1.4	1.429
	80	20	100	RBLR	0.7	114.286	1.4	14.286
50%	5	5	10	FBSR	0.7	7.143	1.4	3.571
	50	50	100	RBLR	0.7	71.429	1.4	35.714

[a] MTBF: Mean Time Between Failures (in hours)
[b] MRT: Mean Repair Time (in hours)
[c] TST: Total System Time (in hours)
[d] FBSR: Frequent breakdown short repair time
[e] RBLR: Rare breakdown long repair time

APPENDIX C. EXPERIMENTAL FACTORS

Table C.1 Designs for serial line experiments

	Design	Line Length	Proc. Time Dist.	Proc. Var.	Workload	Buffer
1	**31111**	3	*LogNormal*	0.3 ***(CV)***	uniform(1)	0
2	**31112**	3	″	″	″	10
3	**31114**	3	″	″	″	100
4	**31121**	3	″	″	bottleneck(1,10%)	0
5	**31122**	3	″	″	″	10
6	**31124**	3	″	″	″	100
7	**31131**	3	″	″	bottleneck(1,20%)	0
8	**31132**	3	″	″	″	10
9	**31134**	3	″	″	″	100
10	**31141**	3	″	″	uniform(0.9)	0
11	**31142**	3	″	″	″	10
12	**31144**	3	″	″	″	100
13	**31151**	3	″	″	bottleneck(0.9,10%)	0
14	**31152**	3	″	″	″	10
15	**31154**	3	″	″	″	100
16	**31161**	3	″	″	bottleneck(0.9,20%)	0
17	**31162**	3	″	″	″	10
18	**31164**	3	″	″	″	100
19	**31171**	3	″	″	uniform(0.5)	0
20	**31172**	3	″	″	″	10
21	**31174**	3	″	″	″	100
22	**31181**	3	″	″	bottleneck(0.5,10%)	0
23	**31182**	3	″	″	″	10
24	**31184**	3	″	″	″	100
25	**31191**	3	″	″	bottleneck(0.5,20%)	0
26	**31192**	3	″	″	″	10
27	**31194**	3	″	″	″	100
28	**311a1**	3	″	″	bottleneck(1,99%)	0
29	**311a2**	3	″	″	″	10
30	**311a4**	3	″	″	″	100
31	**311b1**	3	″	″	bottleneck(0.9,99%)	0
32	**311b2**	3	″	″	″	10
33	**311b4**	3	″	″	″	100
34	**311c1**	3	″	″	bottleneck(0.5,99%)	0
35	**311c2**	3	″	″	″	10
36	**311c4**	3	″	″	″	100
37	**31211**	3	″	2,5 ***(CV)***	uniform(1)	0
38	**31212**	3	″	″	″	10
39	**31214**	3	″	″	″	100
40	**31221**	3	″	″	bottleneck(1,10%)	0
41	**31222**	3	″	″	″	10
42	**31224**	3	″	″	″	100

Table C.1 Designs for serial line experiments (continued)

	Design	Line Length	Proc. Time Dist.	Proc. Var.	Workload	Buffer
43	**31231**	3	*LogNormal*	2,5 ***(CV)***	bottleneck(1,20%)	0
44	**31232**	3	″	″	″	10
45	**31234**	3	″	″	″	100
46	**31241**	3	″	″	uniform(0.9)	0
47	**31242**	3	″	″	″	10
48	**31244**	3	″	″	″	100
49	**31251**	3	″	″	bottleneck(0.9,10%)	0
50	**31252**	3	″	″	″	10
51	**31254**	3	″	″	″	100
52	**31261**	3	″	″	bottleneck(0.9,20%)	0
53	**31262**	3	″	″	″	10
54	**31264**	3	″	″	″	100
55	**31271**	3	″	″	uniform(0.5)	0
56	**31272**	3	″	″	″	10
57	**31274**	3	″	″	″	100
58	**31281**	3	″	″	bottleneck(0.5,10%)	0
59	**31282**	3	″	″	″	10
60	**31284**	3	″	″	″	100
61	**31291**	3	″	″	bottleneck(0.5,20%)	0
62	**31292**	3	″	″	″	10
63	**31294**	3	″	″	″	100
64	**312a1**	3	″	″	bottleneck(1,99%)	0
65	**312a2**	3	″	″	″	10
66	**312a4**	3	″	″	″	100
67	**312b1**	3	″	″	bottleneck(0.9,99%)	0
68	**312b2**	3	″	″	″	10
69	**312b4**	3	″	″	″	100
70	**312c1**	3	″	″	bottleneck(0.5,99%)	0
71	**312c2**	3	″	″	″	10
72	**312c4**	3	″	″	″	100
73	**31611**	3	″	0.3 ***(PV)***	uniform(1)	0
74	**31612**	3	″	″	″	10
75	**31614**	3	″	″	″	100
76	**31621**	3	″	″	bottleneck(1,10%)	0
77	**31622**	3	″	″	″	10
78	**31624**	3	″	″	″	100
79	**31631**	3	″	″	bottleneck(1,20%)	0
80	**31632**	3	″	″	″	10
81	**31634**	3	″	″	″	100
82	**31641**	3	″	″	uniform(0.9)	0
83	**31642**	3	″	″	″	10
84	**31644**	3	″	″	″	100
85	**31651**	3	″	″	bottleneck(0.9,10%)	0
86	**31652**	3	″	″	″	10
87	**31654**	3	″	″	″	100
88	**31661**	3	″	″	bottleneck(0.9,20%)	0

Table C.1 Designs for serial line experiments (continued)

	Design	Line Length	Proc. Time Dist.	Proc. Var.	Workload	Buffer
89	**31662**	3	*Lognormal*	0.3 ***(PV)***	bottleneck(0.9,20%)	10
90	**31664**	3	″	″	″	100
91	**31671**	3	″	″	uniform(0.5)	0
92	**31672**	3	″	″	″	10
93	**31674**	3	″	″	″	100
94	**31681**	3	″	″	bottleneck(0.5,10%)	0
95	**31682**	3	″	″	″	10
96	**31684**	3	″	″	″	100
97	**31691**	3	″	″	bottleneck(0.5,20%)	0
98	**31692**	3	″	″	″	10
99	**31694**	3	″	″	″	100
100	**316a1**	3	″	″	bottleneck(1,99%)	0
101	**316a2**	3	″	″	″	10
102	**316a4**	3	″	″	″	100
103	**316b1**	3	″	″	bottleneck(0.9,99%)	0
104	**316b2**	3	″	″	″	10
105	**316b4**	3	″	″	″	100
106	**316c1**	3	″	″	bottleneck(0.5,99%)	0
107	**316c2**	3	″	″	″	10
108	**316c4**	3	″	″	″	100
109	**31711**	3	″	2,5 ***(PV)***	uniform(1)	0
110	**31712**	3	″	″	″	10
111	**31714**	3	″	″	″	100
112	**31721**	3	″	″	bottleneck(1,10%)	0
113	**31722**	3	″	″	″	10
114	**31724**	3	″	″	″	100
115	**31731**	3	″	″	bottleneck(1,20%)	0
116	**31732**	3	″	″	″	10
117	**31734**	3	″	″	″	100
118	**31741**	3	″	″	uniform(0.9)	0
119	**31742**	3	″	″	″	10
120	**31744**	3	″	″	″	100
121	**31751**	3	″	″	bottleneck(0.9,10%)	0
122	**31752**	3	″	″	″	10
123	**31754**	3	″	″	″	100
124	**31761**	3	″	″	bottleneck(0.9,20%)	0
125	**31762**	3	″	″	″	10
126	**31764**	3	″	″	″	100
127	**31771**	3	″	″	uniform(0.5)	0
128	**31772**	3	″	″	″	10
129	**31774**	3	″	″	″	100
130	**31781**	3	″	″	bottleneck(0.5,10%)	0
131	**31782**	3	″	″	″	10
132	**31784**	3	″	″	″	100
133	**31791**	3	″	″	bottleneck(0.5,20%)	0
134	**31792**	3	″	″	″	10

Table C.1 Designs for serial line experiments (continued)

	Design	Line Length	Proc. Time Dist.	Proc. Var.	Workload	Buffer
135	**31794**	3	*Lognormal*	2.5 ***(PV)***	bottleneck(0.5,20%)	100
136	**317a1**	3	″	″	bottleneck(1,99%)	0
137	**317a2**	3	″	″	″	10
138	**317a4**	3	″	″	″	100
139	**317b1**	3	″	″	bottleneck(0.9,99%)	0
140	**317b2**	3	″	″	″	10
141	**317b4**	3	″	″	″	100
142	**317c1**	3	″	″	bottleneck(0.5,99%)	0
143	**317c2**	3	″	″	″	10
144	**317c4**	3	″	″	″	100
145	**91111**	9	″	0.3 ***(CV)***	uniform(1)	0
146	**91112**	9	″	″	″	10
147	**91114**	9	″	″	″	100
148	**91121**	9	″	″	bottleneck(1,10%)	0
149	**91122**	9	″	″	″	10
150	**91124**	9	″	″	″	100
151	**91131**	9	″	″	bottleneck(1,20%)	0
152	**91132**	9	″	″	″	10
153	**91134**	9	″	″	″	100
154	**911a1**	9	″	″	bottleneck(1,99%)	0
155	**911a2**	9	″	″	″	10
156	**911a4**	9	″	″	″	100
157	**91211**	9	″	2,5 ***(CV)***	uniform(1)	0
158	**91212**	9	″	″	″	10
159	**91214**	9	″	″	″	100
160	**91221**	9	″	″	bottleneck(1,10%)	0
161	**91222**	9	″	″	″	10
162	**91224**	9	″	″	″	100
163	**91231**	9	″	″	bottleneck(1,20%)	0
164	**91232**	9	″	″	″	10
165	**91234**	9	″	″	″	100
166	**912a1**	9	″	″	bottleneck(1,99%)	0
167	**912a2**	9	″	″	″	10
168	**912a4**	9	″	″	″	100

Table C.2 Designs for job-shop experiments

	Design	Line Length	Proc. Time Dist.	Proc. Var.	Workload
1	**31111**	3	*LogNormal*	0.3 ***(CV)***	uniform(80%)
2	**31121**	3	″	″	bottleneck(80%,10%)
3	**31171**	3	″	″	uniform(50%)
4	**31181**	3	″	″	bottleneck(50%,10%)
5	**311x1**	3	″	″	bottleneck(80%,5%)
6	**311y1**	3	″	″	bottleneck(50%,5%)
7	**31211**	3	″	1 ***(CV)***	uniform(80%)
8	**31221**	3	″	″	bottleneck(80%,10%)
9	**31271**	3	″	″	uniform(50%)
10	**31281**	3	″	″	bottleneck(50%,10%)
11	**312x1**	3	″	″	bottleneck(80%,5%)
12	**312y1**	3	″	″	bottleneck(50%,5%)
13	**31611**	3	″	0.3 ***(PV)***	uniform(80%)
14	**31621**	3	″	″	bottleneck(80%,10%)
15	**31671**	3	″	″	uniform(50%)
16	**31681**	3	″	″	bottleneck(50%,10%)
17	**316x1**	3	″	″	bottleneck(80%,5%)
18	**316y1**	3	″	″	bottleneck(50%,5%)
19	**31711**	3	″	1 ***(PV)***	uniform(80%)
20	**31721**	3	″	″	bottleneck(80%,10%)
21	**31771**	3	″	″	uniform(50%)
22	**31781**	3	″	″	bottleneck(50%,10%)
23	**317x1**	3	″	″	bottleneck(80%,5%)
24	**317y1**	3	″	″	bottleneck(50%,5%)
25	**91111**	9	″	0.3 ***(CV)***	uniform(80%)
26	**91171**	9	″	″	uniform(50%)
27	**91211**	9	″	1 ***(CV)***	uniform(80%)
28	**91271**	9	″	″	uniform(50%)

Unreliable design names for both serial line and job-shop systems are given in the following manner:

We include 4 additional digits to the reliable design names to identify the unreliable versions. For example, the unreliable version of design 31111, which is 90% available with frequent breakdown/short repair times is named as 311111221. The additional four digits contain the following information.

Table C.3 Unreliable design names for both serial and job-shop experiments.

Design	Efficiency	Uptime dist.	Downtime Dist.	Breakdown Type
1221	90%	Gamma	Gamma	FBSR[a]
1222	90%	″	″	RBLR[b]
1223	80%	″	″	FBSR
1224	80%	″	″	RBLR
1227	50%	″	″	FBSR
1228	50%	″	″	RBLR

[a] FBSR: Frequent Breakdown/Short Repair Time
[b] RBLR: Rare Breakdown/Long Repair Time

APPENDIX D. SUMMARY STATISTICS

Table D.1 Summary statistics for serial line experiments

Design	Utilization				Hourly		Interdeparture Time (min)	Time in System (min.)
	M/C #1	M/C #2	M/C #3	Average	Throughput	WIP		
31111	0,804	0,804	0,802	0,803	16,9431	2,57	3,54	10,31
31112	0,988	0,988	0,987	0,988	20,8376	13,18	2,88	36,56
31114	1,000	0,999	0,996	0,998	21,0365	59,43	2,85	167,63
31121	0,667	0,926	0,666	0,753	15,6756	2,54	3,83	11,36
31122	0,720	1,000	0,719	0,813	16,9209	12,95	3,55	46,06
31124	0,722	1,000	0,719	0,814	16,9209	102,56	3,55	363,09
31131	0,506	0,979	0,506	0,664	13,2483	2,52	4,53	13,55
31132	0,517	1,000	0,516	0,678	13,5333	12,68	4,33	57,61
31134	0,518	1,000	0,516	0,678	13,5333	102,51	4,33	405,09
31141	0,820	0,820	0,819	0,820	19,2757	2,58	3,11	9,09
31142	0,991	0,990	0,989	0,990	23,2814	13,11	2,58	32,52
31144	1,000	0,999	0,997	0,999	23,4144	54,59	2,56	138,26
31151	0,695	0,932	0,694	0,774	17,9949	2,54	3,33	9,90
31152	0,746	1,000	0,745	0,830	19,3166	12,96	3,11	40,35
31154	0,748	1,000	0,745	0,831	19,3166	102,51	3,11	317,89
31161	0,544	0,980	0,543	0,689	15,5116	2,52	3,87	11,57
31162	0,555	1,000	0,554	0,703	15,8320	12,68	3,79	49,25
31164	0,557	1,000	0,554	0,704	15,8320	102,48	3,79	388,92
31171	0,893	0,893	0,892	0,893	32,1090	2,58	1,87	5,52
31172	0,997	0,997	0,996	0,997	35,8640	12,74	1,67	20,85
31174	1,000	0,999	0,998	0,999	35,9282	30,94	1,67	50,76
31181	0,817	0,957	0,817	0,864	30,9820	2,5384	1,9364	5,7744
31182	0,855	1,000	0,854	0,903	32,3849	12,97	1,8526	24,0623
31184	0,857	1,000	0,854	0,904	32,3849	102,04	1,85	188,72
31191	0,719	0,986	0,719	0,808	28,7112	2,52	2,09	6,26
31192	0,730	1,000	0,729	0,820	29,1286	12,68	2,06	26,76
31194	0,732	1,000	0,729	0,820	29,1286	102,25	2,06	210,87
311a1	0,034	1,000	0,034	0,356	2,0396	2,50	29,42	88,24
311a2	0,034	1,000	0,034	0,356	2,0396	12,50	29,42	382,31
311a4	0,034	1,000	0,034	0,356	2,0396	102,49	29,42	3024,54
311b1	0,050	1,000	0,050	0,367	2,9648	2,50	20,24	62,54
311b2	0,050	1,000	0,050	0,367	2,9648	12,50	20,24	262,99
311b4	0,050	1,000	0,050	0,367	2,9648	102,49	20,24	2080,58
311c1	0,205	1,000	0,205	0,470	12,2232	2,50	4,91	14,72
311c2	0,205	1,000	0,205	0,470	12,2233	12,50	4,91	63,79
311c4	0,205	1,000	0,205	0,470	12,2233	102,46	4,91	504,49
31211	0,358	0,386	0,347	0,364	1,0003	2,36	163,33	437,31
31212	0,456	0,491	0,442	0,463	1,0005	12,88	128,25	1620,03
31214	0,647	0,685	0,615	0,649	1,0010	104,32	92,16	9414,87
31221	0,086	0,868	0,084	0,346	1,0002	2,37	353,01	1035,06
31222	0,094	0,946	0,092	0,377	1,0002	12,82	324,35	4203,28
31224	0,010	0,997	0,097	0,368	1,0002	102,28	308,1	31656,79
31231	0,009	0,988	0,009	0,335	1,0001	2,39	1891,55	5662,97
31232	0,009	0,999	0,009	0,339	1,0001	12,77	1873,40	24342,96
31234	0,009	1,000	0,009	0,339	1,0001	102,70	1870,61	192468,00
31241	0,370	0,386	0,347	0,368	1,0004	2,36	147,79	395,69
31242	0,456	0,491	0,442	0,463	1,0006	12,88	116,04	1465,86
31244	0,715	0,746	0,694	0,718	1,4247	104,21	42,02	4299,60

Table D.1 Summary statistics for serial line experiments (continued)

Design	Utilization				Hourly		Interdeparture Time (min)	Time in System (min.)
	M/C #1	M/C #2	M/C #3	Average	Throughput	WIP		
31251	0,123	0,823	0,121	0,356	1,0003	2,38	139,89	406,94
31252	0,141	0,938	0,138	0,406	1,0008	12,85	122,89	1591,97
31254	0,150	0,998	0,147	0,432	1,0015	102,35	115,58	11870,88
31261	0,020	0,976	0,020	0,339	1,0002	2,39	529,23	1581,15
31262	0,020	0,997	0,020	0,346	1,0002	12,79	518,13	6731,74
31264	0,020	1,000	0,020	0,347	1,0002	102,72	516,89	53167,00
31271	0,490	0,491	0,485	0,489	8,1543	2,39	7,36	20,19
31272	0,782	0,783	0,773	0,779	13,0001	13,29	4,61	59,93
31274	0,962	0,961	0,948	0,957	15,9501	100,05	3,76	372,42
31281	0,344	0,655	0,394	0,464	6,9738	2,41	8,6	25,02
31282	0,497	0,946	0,570	0,671	10,0771	13,96	5,95	82,66
31284	0,527	1,000	0,520	0,682	10,6473	102,74	5,63	576,31
31291	0,221	0,839	0,219	0,426	5,3779	2,43	11,15	32,6
31292	0,263	0,997	0,261	0,507	6,3888	12,95	9,38	121,88
31294	0,264	1,000	0,262	0,509	6,4099	102,86	9,36	961,39
312a1	0,000	1,000	0,000	0,333	1,0000	2,75	1,02E+11	3,05E+11
312a2	0,000	1,000	0,000	0,333	1,0000	12,88	1,02E+11	1,32E+12
312a4	0,000	1,000	0,000	0,333	1,0000	102,88	1,02E+11	1,05E+13
312b1	0,000	1,000	0,000	0,333	1,0000	2,69	2,86E+09	8,58E+09
312b2	0,000	1,000	0,000	0,333	1,0000	12,85	2,86E+09	3,72E+10
312b4	0,000	1,000	0,000	0,333	1,0000	102,85	2,86E+09	2,95E+11
312c1	0,000	1,000	0,000	0,333	1,0001	2,55	4,48E+03	1,34E+04
312c2	0,000	1,000	0,000	0,333	1,0001	12,87	4,48E+03	5,83E+04
312c4	0,000	1,000	0,000	0,333	1,0001	102,87	4,48E+03	4,61E+05
31611	0,804	0,804	0,802	0,803	16,9431	2,57	3,54	10,31
31612	0,988	0,988	0,987	0,988	20,8376	13,18	2,88	36,56
31614	1,000	0,999	0,996	0,998	21,0365	59,43	2,85	167,63
31621	0,684	0,923	0,683	0,763	15,9394	2,53	3,76	11,16
31622	0,741	1,000	0,740	0,827	17,2627	12,96	3,48	45,15
31624	0,743	1,000	0,740	0,828	18,4529	94,16	3,29	315,61
31631	0,538	0,981	0,537	0,685	13,8631	2,51	4,33	12,94
31632	0,549	1,000	0,548	0,699	14,1341	12,65	4,24	55,15
31634	0,551	1,000	0,548	0,700	14,1341	102,46	4,24	435,63
31641	0,804	0,804	0,802	0,803	18,7243	2,57	3,20	9,33
31642	0,988	0,988	0,987	0,988	23,0297	13,18	2,61	33,08
31644	1,000	0,999	0,996	0,998	23,2492	59,43	2,58	151,67
31651	0,698	0,914	0,697	0,770	17,7948	2,54	3,37	9,98
31652	0,764	1,000	0,762	0,842	19,4636	13,01	3,08	40,04
31654	0,766	1,000	0,762	0,843	19,4636	102,51	3,08	315,37
31661	0,567	0,974	0,566	0,702	15,8299	2,52	3,79	11,32
31662	0,583	1,000	0,582	0,722	16,2586	12,69	3,69	47,95
31664	0,585	1,000	0,582	0,722	16,2586	102,47	3,69	378,61
31671	0,804	0,804	0,802	0,803	27,9338	2,57	2,15	6,25
31672	0,988	0,988	0,987	0,988	34,3552	13,18	1,75	22,18
31674	1,000	0,999	0,996	0,998	34,6828	59,43	1,73	101,67
31681	0,750	0,871	0,748	0,790	27,3905	2,55	2,19	6,44
31682	0,861	1,000	0,859	0,907	31,4545	13,21	1,91	24,77
31684	0,863	1,000	0,859	0,907	31,4545	102,26	1,91	194,28
31691	0,684	0,923	0,683	0,763	26,2781	2,53	2,28	6,77
31692	0,741	1,000	0,740	0,827	28,4615	12,96	2,11	27,39
31694	0,743	1,000	0,740	0,828	28,4615	102,52	2,11	215,79
316a1	0,051	1,000	0,051	0,367	2,9116	2,5	20,61	61,82
316a2	0,051	1,000	0,051	0,367	2,9116	12,5	20,61	267,84

Table D.1 Summary statistics for serial line experiments (continued)

Design	Utilization				Hourly		Interdeparture Time (min)	Time in System (min.)
	M/C #1	M/C #2	M/C #3	Average	Throughput	WIP		
316a4	0,051	1,000	0,051	0,367	2,9116	102,49	20,61	2118,85
316b1	0,069	1,000	0,069	0,379	3,9223	2,5	15,3	45,89
316b2	0,069	1,000	0,069	0,379	3,9223	12,5	15,29	198,82
316b4	0,069	1,000	0,069	0,379	3,9223	102,49	15,29	1572,78
316c1	0,226	1,000	0,226	0,484	12,9178	2,5	4,64	13,93
316c2	0,227	1,000	0,226	0,484	12,9179	12,5	4,64	60,36
316c4	0,227	1,000	0,226	0,484	12,9179	102,46	4,64	477,35
31711	0,358	0,386	0,347	0,364	1,0003	2,36	163,33	437,31
31712	0,456	0,491	0,442	0,463	1,0005	12,88	128,25	1620,03
31714	0,637	0,685	0,615	0,646	1,0010	104,32	92,16	9414,87
31721	0,317	0,462	0,308	0,362	1,0003	2,36	166,62	452,76
31722	0,4	0,582	0,388	0,457	1,0004	12,86	132,09	1677,43
31724	0,549	0,796	0,531	0,625	1,0010	103,01	96,73	9878,83
31731	0,275	0,540	0,267	0,361	1,0002	2,37	173,98	479,89
31732	0,342	0,670	0,331	0,448	1,0004	12,82	140,12	1787,73
31734	0,450	0,878	0,435	0,588	1,0007	102,65	107,08	10928,00
31741	0,358	0,387	0,347	0,364	1,0004	2,36	147,79	395,69
31742	0,456	0,491	0,442	0,463	1,0006	12,88	116,04	1465,86
31744	0,637	0,685	0,615	0,646	1,0022	104,32	83,39	8518,92
31751	0,322	0,454	0,312	0,363	1,0003	2,36	150,31	407,82
31752	0,407	0,573	0,394	0,458	1,0005	12,86	119,01	1510,53
31754	0,560	0,787	0,541	0,629	1,0022	103,14	86,79	8863,80
31761	0,284	0,524	0,275	0,361	1,0003	2,37	155,77	428,40
31762	0,354	0,653	0,343	0,450	1,0004	12,83	125,01	1593,55
31764	0,497	0,864	0,454	0,605	1,0012	102,88	94,60	9661,74
31771	0,358	0,386	0,347	0,364	1,0006	2,36	99,07	265,24
31772	0,456	0,491	0,442	0,463	1,0022	12,88	77,79	982,60
31774	0,637	0,685	0,615	0,646	1,0961	104,32	55,89	5710,41
31781	0,338	0,423	0,328	0,363	1,0006	2,36	99,77	269,10
31782	0,429	0,537	0,416	0,461	1,0021	12,87	78,64	996,00
31784	0,643	0,744	0,576	0,654	1,0866	103,92	56,81	5810,94
31791	0,317	0,462	0,308	0,362	1,0006	2,36	101,06	274,61
31792	0,400	0,582	0,388	0,457	1,0020	12,86	80,12	1017,42
31794	0,549	0,796	0,531	0,625	1,0655	103,01	58,67	5991,81
317a1	0,045	0,939	0,043	0,342	1,0000	2,43	485,86	1441,81
317a2	0,047	0,984	0,046	0,359	1,0000	12,62	463,63	6016,95
317a4	0,048	0,999	0,046	0,364	1,0000	102,56	456,65	46944,00
317b1	0,063	0,909	0,062	0,345	1,0001	2,42	372,28	1099,01
317b2	0,068	0,971	0,066	0,368	1,0001	12,66	348,70	4522,67
317b4	0,070	0,998	0,068	0,379	1,0001	102,53	339,32	34876,00
317c1	0,158	0,750	0,153	0,354	1,0003	2,39	137,03	393,09
317c2	0,183	0,868	0,179	0,410	1,0005	12,74	118,37	1527,01
317c4	0,208	0,980	0,201	0,463	1,0009	102,33	105,01	10766,00
311111221	0,595	0,595	0,594	0,595	12,5504	2,56	4,78	13,58
311121221	0,764	0,764	0,763	0,764	16,1300	13,67	3,72	48,89
311141221	0,865	0,863	0,860	0,863	18,1700	102,11	3,30	332,57
311211221	0,495	0,687	0,494	0,559	11,6313	2,53	5,16	14,89
311221221	0,585	0,813	0,584	0,661	13,7486	13,03	4,36	56,21
311241221	0,653	0,905	0,651	0,736	15,3106	101,59	3,92	397,02
311311221	0,378	0,731	0,377	0,495	9,8861	2,50	6,06	17,65
311321221	0,430	0,832	0,429	0,564	11,2486	12,72	5,33	68,95
311341221	0,469	0,905	0,467	0,614	12,2470	101,89	4,90	499,91

Table D.1 Summary statistics for serial line experiments (continued)

Design	Utilization				Hourly		Interdeparture Time (min)	Time in System (min.)
	M/C #1	M/C #2	M/C #3	Average	Throughput	WIP		
311411221	0,607	0,607	0,606	0,607	14,2788	2,57	4,20	11,96
311421221	0,764	0,764	0,763	0,764	17,9171	13,55	3,34	43,48
311441221	0,863	0,860	0,858	0,860	20,1797	98,94	2,97	289,71
311511221	0,512	0,689	0,511	0,571	13,2315	2,53	4,53	13,06
311521221	0,599	0,806	0,598	0,668	15,5002	13,07	3,87	50,11
311541221	0,674	0,905	0,671	0,750	17,4047	101,71	3,45	349,52
311611221	0,404	0,728	0,404	0,512	11,5355	2,50	5,20	15,12
311621221	0,457	0,825	0,456	0,579	13,0104	12,73	4,61	59,43
311641221	0,504	0,905	0,502	0,637	14,3500	101,64	4,19	425,98
311711221	0,654	0,654	0,653	0,654	23,5022	2,58	2,55	7,30
311721221	0,755	0,755	0,754	0,755	27,1359	13,51	2,21	28,43
311741221	0,847	0,846	0,843	0,845	30,3062	101,31	1,98	197,72
311811221	0,600	0,702	0,599	0,634	22,7123	2,53	2,64	7,61
311821221	0,663	0,776	0,663	0,701	25,1200	13,07	2,39	30,64
311841221	0,757	0,883	0,754	0,798	28,5771	100,62	2,09	209,49
311911221	0,529	0,725	0,529	0,594	21,0874	2,51	2,84	8,21
311921221	0,572	0,784	0,572	0,643	22,8327	12,76	2,62	33,65
311941221	0,654	0,894	0,651	0,733	25,9979	100,87	2,30	231,54
311a11221	0,028	0,827	0,028	0,294	1,6863	2,43	35,58	105,35
311a21221	0,031	0,901	0,031	0,321	1,8373	12,42	32,66	424,59
311a41221	0,031	0,901	0,031	0,321	1,8384	102,42	32,64	3355,92
311b11221	0,040	0,806	0,040	0,295	2,3911	2,44	25,09	74,02
311b21221	0,045	0,900	0,045	0,330	2,6686	12,43	22,48	292,25
311b41221	0,045	0,903	0,045	0,331	2,6762	102,43	22,42	2305,41
311c11221	0,154	0,753	0,154	0,354	9,2183	2,48	6,52	19,04
311c21221	0,173	0,845	0,173	0,397	10,3355	12,47	5,81	75,28
311c41221	0,186	0,903	0,185	0,425	11,0431	102,31	5,43	557,83
312111221	0,327	0,316	0,315	0,319	1,0002	2,32	185,18	496,26
312121221	0,420	0,406	0,404	0,410	1,0003	12,83	143,14	1809,22
312141221	0,593	0,572	0,569	0,578	1,0006	104,03	102,28	10448,1
312211221	0,086	0,757	0,084	0,309	1,0001	2,33	395,92	1159,88
312221221	0,096	0,841	0,093	0,343	1,0002	12,80	360,66	4674,08
312241221	0,102	0,896	0,100	0,366	1,0002	102,25	342,18	35166
312311221	0,010	0,885	0,010	0,302	1,0001	2,34	2105,16	6301,17
312321221	0,010	0,898	0,010	0,306	1,0001	12,71	2081,60	27047,96
312341221	0,010	0,900	0,010	0,307	1,0001	102,66	2078,51	213833,00
312411221	0,329	0,324	0,322	0,325	1,0004	2,33	92,40	248,13
312421221	0,450	0,444	0,440	0,445	1,0045	12,89	66,84	847,20
312441221	0,651	0,640	0,635	0,642	1,2825	104,23	46,77	4784,65
312511221	0,116	0,713	0,114	0,314	1,0002	2,34	171,79	513,49
312521221	0,136	0,834	0,133	0,368	1,0003	13,41	147,24	2050,32
312541221	0,146	0,898	0,144	0,396	1,0007	11,97	132,88	14767,49
312611221	0,020	0,870	0,020	0,303	1,0001	2,35	591,09	1764,65
312621221	0,021	0,897	0,021	0,313	1,0002	12,76	575,78	7481,11
312641221	0,021	0,900	0,021	0,314	1,0002	102,62	574,12	59064,62
312711221	0,376	0,377	0,373	0,375	6,2485	2,38	9,6	26,21
312721221	0,633	0,635	0,627	0,632	10,5342	13,23	5,69	73,69
312741221	0,844	0,844	0,833	0,840	13,9192	102,69	4,31	436,66
312811221	0,280	0,533	0,278	0,364	5,6514	2,39	10,61	29,83
312821221	0,421	0,800	0,417	0,546	8,5561	13,04	7,02	90,81
312841221	0,476	0,902	0,470	0,616	9,6247	102,84	6,22	637,94
312911221	0,176	0,668	0,175	0,340	4,2826	2,41	13,99	40,29

Table D.1 Summary statistics for serial line experiments (continued)

Design	Utilization				Hourly		Interdeparture Time (min)	Time in System (min.)
	M/C #1	M/C #2	M/C #3	Average	Throughput	WIP		
312921221	0,228	0,870	0,226	0,441	5,5333	12,89	10,84	140,9
312941221	0,240	0,906	0,237	0,461	5,8086	102,78	10,33	1061,04
312c11221	0,000	0,899	0,000	0,300	1,0001	2,49	4983,59	14947,2
312c21221	0,000	0,900	0,000	0,300	1,0001	12,74	4980,67	64732,7
312c41221	0,000	0,900	0,000	0,300	1,0001	102,72	4980,71	512570,80
316111221	0,595	0,595	0,594	0,595	12,5500	2,56	4,78	13,58
316121221	0,764	0,764	0,763	0,764	16,1300	13,67	3,72	48,89
316141221	0,865	0,863	0,860	0,863	18,1700	102,1	3,3	332,57
316211221	0,507	0,685	0,507	0,566	11,8106	2,52	5,07	14,63
316221221	0,602	0,812	0,601	0,672	14,0227	13,04	4,28	55,19
316241221	0,673	0,905	0,654	0,744	15,6323	101,53	3,84	388,79
316311221	0,401	0,730	0,400	0,510	10,3330	2,49	5,81	16,91
316321221	0,455	0,828	0,454	0,579	11,7082	12,71	5,12	66,22
316341221	0,498	0,905	0,496	0,633	12,7985	101,75	4,69	477,74
316411221	0,595	0,595	0,594	0,595	13,8572	2,56	4,33	12,29
316421221	0,763	0,763	0,762	0,763	17,7526	12,51	4,37	43,81
316441221	0,862	0,860	0,857	0,860	19,9989	99,33	2,99	293,63
316511221	0,517	0,677	0,516	0,570	13,1847	2,53	4,55	13,09
316521221	0,615	0,805	0,613	0,678	15,6634	13,09	3,83	49,55
316541221	0,692	0,904	0,688	0,761	17,5866	101,76	3,41	345,81
316611221	0,421	0,723	0,421	0,522	11,7486	2,50	5,10	14,80
316621221	0,477	0,818	0,476	0,590	13,3087	12,74	4,51	58,08
316641221	0,529	0,906	0,527	0,654	14,7265	101,69	4,07	414,73
316711221	0,593	0,593	0,592	0,593	20,5912	2,56	2,91	8,25
316721221	0,748	0,748	0,746	0,747	25,9762	13,59	2,31	29,91
316741221	0,847	0,846	0,844	0,846	29,3532	101,84	2,04	205,43
316811221	0,554	0,642	0,552	0,583	20,2056	2,54	2,97	8,48
316821221	0,666	0,773	0,665	0,701	24,3437	13,29	2,46	31,77
316841221	0,762	0,884	0,759	0,802	27,7951	99,97	2,16	213,89
316911221	0,504	0,680	0,503	0,562	19,3496	2,53	3,09	8,92
316921221	0,581	0,784	0,580	0,648	22,3366	12,98	2,69	34,44
316941221	0,665	0,895	0,662	0,741	25,4379	101,09	2,35	236,97
316a11221	0,041	0,803	0,041	0,295	2,3355	2,43	26,68	75,72
316a21221	0,046	0,901	0,046	0,331	2,6229	12,43	22,87	297,35
316a41221	0,047	0,902	0,046	0,332	2,6268	102,42	22,84	2348,16
316b11221	0,040	0,806	0,040	0,295	2,3911	2,44	25,09	74,02
316b21221	0,045	0,900	0,045	0,330	2,6686	12,43	22,48	292,25
316b41221	0,045	0,903	0,045	0,331	2,6762	102,43	22,42	2305,42
316c11221	0,170	0,752	0,170	0,364	9,7059	2,48	6,18	18,03
316c21221	0,191	0,841	0,190	0,407	10,8615	12,48	5,52	71,45
316c41221	0,205	0,903	0,204	0,437	11,6769	102,26	5,14	527,74
317111221	0,316	0,340	0,306	0,321	1,0002	2,32	185,18	496,26
317121221	0,408	0,440	0,396	0,415	1,0003	12,83	143,14	1809,22
317141221	0,573	0,617	0,554	0,581	1,0006	104,03	102,28	10448,15
317211221	0,280	0,408	0,272	0,320	1,0002	2,33	188,72	512,68
317221221	0,359	0,522	0,348	0,410	1,0003	12,84	147,43	1872,36
317241221	0,495	0,717	0,478	0,563	1,0006	103,04	107,24	10957,34
317311221	0,243	0,477	0,235	0,318	1,0001	2,33	197,00	543,24
317321221	0,307	0,601	0,267	0,392	1,0002	12,80	156,28	1994,96
317341221	0,405	0,791	0,391	0,529	1,0004	102,51	118,97	12136,43
317411221	0,315	0,340	0,305	0,320	1,0002	2,32	167,72	449,21
317421221	0,408	0,440	0,395	0,414	1,0004	12,87	129,63	1638,95

Table D.1 Summary statistics for serial line experiments (continued)

Design	Utilization				Hourly		Interdeparture Time (min)	Time in System (min.)
	M/C #1	M/C #2	M/C #3	Average	Throughput	WIP		
317441221	0,573	0,616	0,554	0,581	1,0010	104,84	92,69	9502,16
317511221	0,283	0,399	0,274	0,319	1,0002	2,33	170,69	463,35
317521221	0,364	0,513	0,352	0,410	1,0004	12,85	132,99	1687,85
317541221	0,503	0,708	0,487	0,566	1,0010	103,05	96,47	9837,89
317611221	0,250	0,462	0,242	0,318	1,0002	2,33	176,79	486,06
317621221	0,316	0,585	0,307	0,403	1,0003	12,81	139,69	1780,06
317641221	0,422	0,777	0,408	0,536	1,0007	103,01	105,21	10749,22
317711221	0,313	0,338	0,303	0,318	1,0004	2,32	113,22	303,29
317721221	0,407	0,438	0,394	0,413	1,0012	12,85	87,17	1101,74
317741221	0,573	0,617	0,555	0,582	1,0279	104,32	62,05	6336,62
317811221	0,296	0,371	0,287	0,318	1,0004	2,33	113,88	306,99
317821221	0,383	0,479	0,371	0,411	1,0011	12,87	88,05	1117,67
317841221	0,536	0,669	0,519	0,575	1,0259	103,91	63,13	6461,38
317911221	0,278	0,404	0,269	0,317	1,0004	2,33	115,39	313,54
317921221	0,358	0,521	0,347	0,409	1,0010	12,82	89,61	1137,77
317941221	0,494	0,716	0,478	0,563	1,0180	103,11	65,19	6660,17
317a11221	0,040	0,839	0,039	0,306	1,0000	2,37	543,53	1611,56
317a21221	0,042	0,886	0,041	0,323	1,0000	12,58	515,18	6685,70
317a41221	0,043	0,899	0,042	0,328	1,0000	102,49	507,26	52148,60
317b11221	0,053	0,818	0,051	0,307	1,0000	2,36	413,89	1222,29
317b21221	0,056	0,876	0,055	0,329	1,0000	12,59	386,29	5011,10
317b41221	0,058	0,899	0,056	0,338	1,0000	102,49	376,79	37734,05
317c11221	0,139	0,660	0,135	0,311	1,0002	2,35	155,63	445,69
317c21221	0,165	0,779	0,159	0,368	1,0003	12,69	131,86	1702,20
317c41221	0,187	0,881	0,180	0,416	1,0005	102,51	116,68	11969,52
311111222	0,589	0,589	0,588	0,589	12,74	2,57	4,68	13,31
311121222	0,714	0,714	0,712	0,713	15,63	13,16	3,81	48,18
311141222	0,759	0,760	0,757	0,759	16,34	92,53	3,63	341,41
311211222	0,475	0,660	0,474	0,536	11,76	2,53	5,12	14,64
311221222	0,533	0,740	0,532	0,602	12,88	12,95	4,63	59,56
311241222	0,572	0,795	0,571	0,646	13,97	103,63	4,28	439,84
311311222	0,350	0,678	0,350	0,459	9,82	2,51	6,12	17,58
311321222	0,368	0,712	0,367	0,482	10,21	12,66	5,87	74,08
311341222	0,414	0,800	0,413	0,542	11,31	102,13	5,29	533,48
311a11222	0,026	0,749	0,026	0,267	1,53	2,49	39,11	114,05
311a21222	0,028	0,814	0,028	0,290	1,67	12,46	36,01	465,61
311a41222	0,031	0,900	0,031	0,321	1,85	102,25	32,42	3330,29
312111222	0,310	0,300	0,298	0,303	1,00	2,34	196,03	524,51
312121222	0,410	0,396	0,394	0,400	1,00	12,86	147,62	1866,08
312141222	0,59	0,569	0,566	0,575	1,0006	104,27	103,05	10549,9
312211222	0,083	0,733	0,081	0,299	1,00	2,35	406,91	1188,19
312221222	0,094	0,829	0,092	0,338	1,00	12,76	365,14	4729,43
312241222	0,102	0,895	0,099	0,365	1,0002	102,27	341,68	35113,21
312311222	0,01	0,88	0,01	0,300	1,0001	2,36	2117,13	6332,11
312321222	0,01	0,898	0,01	0,306	1,0001	12,69	2084,39	27085,1
312341222	0,010	0,902	0,010	0,307	1,00	102,54	2077,56	213758,00
311111223	0,420	0,420	0,419	0,420	8,91	2,56	6,76	18,88
311121223	0,553	0,553	0,552	0,553	11,66	13,49	5,14	66,06
311141223	0,703	0,703	0,701	0,702	14,87	101,13	4,02	404,33
311211223	0,279	0,270	0,269	0,273	1,00	2,29	217,16	582,21
311221223	0,369	0,357	0,355	0,360	1,00	12,88	163,43	2068,53
311241223	0,524	0,506	0,503	0,511	1,00	104,19	115,60	11826,87

Table D.1 Summary statistics for serial line experiments (continued)

Design	Utilization				Hourly		Interdeparture Time (min)	Time in System (min.)
	M/C #1	M/C #2	M/C #3	Average	Throughput	WIP		
311111224	0,401	0,401	0,400	0,401	8,95	2,57	6,71	18,50
311121224	0,515	0,515	0,513	0,514	11,52	13,16	5,17	64,15
311141224	0,545	0,545	0,543	0,544	12,48	92,79	4,76	449,82
311211224	0,242	0,234	0,233	0,236	1,00	2,33	250,41	670,91
311221224	0,332	0,321	0,320	0,324	1,00	12,81	181,15	2283,67
311241224	0,518	0,500	0,498	0,505	1,00	103,41	118,72	12104,44
311111227	0,106	0,106	0,106	0,106	2,25	2,52	26,49	71,71
311121227	0,167	0,167	0,166	0,167	3,45	8,83	17,35	219,47
311141227	0,320	0,320	0,319	0,320	6,80	103,05	13,20	911,74
311211227	0,142	0,138	0,137	0,139	1,00	2,16	426,43	1147,21
311221227	0,218	0,211	0,210	0,213	1,00	12,73	276,58	3509,29
311241227	0,326	0,314	0,313	0,318	1,00	104,83	186,65	19228,74
311111228	0,099	0,099	0,099	0,099	2,20	2,56	28,06	73,67
311121228	0,123	0,123	0,123	0,123	2,80	13,30	21,02	253,91
311141228	0,149	0,149	0,149	0,149	3,42	98,16	17,15	1612,76
311211228	0,084	0,081	0,081	0,082	1,00	2,27	723,84	1935,32
311221228	0,149	0,145	0,144	0,146	1,00	12,66	404,07	5120,72
311241228	0,285	0,275	0,274	0,278	1,00	100,34	211,51	21202,54

Table D.1 Summary statistics for serial line experiments (continued)

Design	Utilization										Hourly		Interdeparture Time (min)	Time in System (min.)
	M/C #1	M/C #2	M/C #3	M/C #4	M/C #5	M/C #6	M/C #7	M/C #8	M/C #9	Average	Throughput	WIP		
91111	0,727	0,727	0,725	0,726	0,729	0,726	0,727	0,723	0,726	0,726	15,3134	8,15	3,92	32,13
91112	0,984	0,984	0,982	0,983	0,987	0,982	0,983	0,978	0,982	0,983	20,7109	51,01	2,90	146,21
91114	1,000	0,999	0,997	0,996	0,997	0,992	0,993	0,988	0,992	0,995	20,9176	168,16	2,87	476,94
91121	0,357	0,357	0,356	0,357	0,988	0,357	0,357	0,355	0,357	0,427	8,4508	6,47	7,10	50,34
91122	0,362	0,362	0,361	0,361	1,000	0,361	0,361	0,360	0,361	0,432	8,5516	46,50	7,02	271,34
91124	0,366	0,366	0,364	0,363	1,000	0,361	0,361	0,360	0,361	0,434	8,5392	385,01	7,03	2639,37
91131	0,124	0,124	0,123	0,123	0,999	0,123	0,124	0,123	0,123	0,221	3,2740	5,45	18,33	116,88
91132	0,124	0,124	0,124	0,124	1,000	0,124	0,124	0,123	0,124	0,221	3,2764	45,45	18,32	707,50
91134	0,125	0,125	0,125	0,124	1,000	0,124	0,124	0,123	0,124	0,222	3,2692	384,92	18,36	6926,34
911a1	0,000	0,000	0,000	0,000	1,000	0,000	0,000	0,000	0,000	0,111	1,0000	5,00	265272,30	1,59E+06
911a2	0,000	0,000	0,000	0,000	1,000	0,000	0,000	0,000	0,000	0,111	1,0000	45,00	265272,30	1,14E+07
911a4	0,000	0,000	0,000	0,000	1,000	0,000	0,000	0,000	0,000	0,111	1,0000	405,00	265272,30	1,06E+08
91211	0,153	0,148	0,147	0,172	0,168	0,160	0,145	0,138	0,142	0,153	1,0000	5,56	393,67	2401,37
91212	0,250	0,242	0,241	0,281	0,274	0,261	0,236	0,225	0,231	0,249	1,0000	48,94	237,39	11523,55
91214	0,449	0,434	0,435	0,501	0,491	0,464	0,420	0,400	0,411	0,445	1,0000	429,89	133,93	57648,83
91221	0,000	0,000	0,000	0,000	0,999	0,000	0,000	0,000	0,000	0,111	1,0000	5,51	81509,53	3764,45
91222	0,000	0,000	0,000	0,000	1,000	0,000	0,000	0,000	0,000	0,111	1,0000	47,24	81411,75	3,74E+06
91224	0,000	0,000	0,000	0,000	1,000	0,000	0,000	0,000	0,000	0,111	1,0000	387,50	80574,04	2,58E+07
91231	0,000	0,000	0,000	0,000	1,000	0,000	0,000	0,000	0,000	0,111	1,0000	5,46	5,21E+08	3,13E+09
91232	0,000	0,000	0,000	0,000	1,000	0,000	0,000	0,000	0,000	0,111	1,0000	46,26	5,21E+08	2,40E+10
91234	0,000	0,000	0,000	0,000	1,000	0,000	0,000	0,000	0,000	0,111	1,0000	386,05	5,14E+08	1,56E+11
912a1	0,000	0,000	0,000	0,000	1,000	0,000	0,000	0,000	0,000	0,111	1,0000	8,5717	3,92E+32	2,35E+33
912a2	0,000	0,000	0,000	0,000	1,000	0,000	0,000	0,000	0,000	0,111	1,0000	48,5815	3,92E+32	1,68E+34
912a4	0,000	0,000	0,000	0,000	1,000	0,000	0,000	0,000	0,000	0,111	1,0000	408,58	3,92E+32	1,39E+35
911111222	0,259	0,259	0,259	0,259	0,260	0,259	0,259	0,258	0,259	0,259	5,9170	8,0593	10,165	53,7871
911121222	0,412	0,412	0,411	0,411	0,413	0,411	0,411	0,409	0,411	0,411	8,9942	50,3803	6,7852	321,4086
911141222	0,608	0,606	0,605	0,605	0,608	0,604	0,604	0,599	0,601	0,604	12,5941	442,02	4,70	2084,67
911211222	0,134	0,134	0,134	0,134	0,134	0,371	0,134	0,134	0,134	0,160	3,3981	6,41	17,74	112,32
911221222	0,184	0,184	0,184	0,184	0,508	0,183	0,184	0,183	0,183	0,220	4,5157	46,40	13,28	616,99
911241222	0,311	0,310	0,309	0,308	0,850	0,307	0,307	0,306	0,307	0,368	7,0650	430,89	8,42	3632,73
912111222	0,118	0,115	0,114	0,133	0,130	0,124	0,112	0,107	0,110	0,118	1,0000	5,4731	508,1284	3106,87
912121222	0,217	0,210	0,209	0,244	0,238	0,227	0,205	0,195	0,201	0,216	1,0000	49,1238	272,93	13301,35
912141222	0,403	0,389	0,389	0,449	0,440	0,416	0,377	0,359	0,369	0,399	1,0000	436,26	151,40	65300,49
912211222	0,000	0,000	0,000	0,000	0,899	0,000	0,000	0,000	0,000	0,100	1,0000	5,39	90631,14	543771,90
912221222	0,000	0,000	0,000	0,000	0,900	0,000	0,000	0,000	0,000	0,100	1,0000	47,45	79180,89	3,64E+06
912241222	0,000	0,000	0,000	0,000	0,900	0,000	0,000	0,000	0,000	0,100	1,0000	408,28	107068,63	3,50E+07

Table D.2 Summary statistics for job-shop experiments

Design	Utilization				Hourly		Interdeparture Time (min)	Time in System (min.)
	M/C #1	M/C #2	M/C #3	Average	Throughput	WIP		
31111	0,800	0,790	0,799	0,796	16,9128	7,35	3,5466	22,97
31121	0,596	0,925	0,595	0,705	16,8785	77,54	3,5545	271,75
31171	0,502	0,501	0,502	0,502	10,6281	3,14	5,64	12,53
31181	0,380	0,712	0,378	0,490	10,6280	3,31	5,64	13,59
311x1	0,702	0,856	0,703	0,754	16,9118	8,65	3,55	27,72
311y1	0,423	0,626	0,425	0,491	10,6280	3,18	5,64	12,75
31211	0,807	0,799	0,809	0,805	10,8133	18,19	5,55	94,12
31221	0,587	0,942	0,586	0,705	8,8800	1838,03	6,76	10201,88
31271	0,504	0,499	0,504	0,502	6,7469	4,85	8,89	32,76
31281	0,362	0,724	0,358	0,481	6,7456	8,71	8,89	64,42
312x1	0,700	0,864	0,701	0,755	10,8087	82,08	5,55	445,75
312y1	0,418	0,635	0,417	0,490	6,7468	5,43	8,89	37,37
31611	0,800	0,798	0,799	0,799	16,9128	7,35	3,55	22,97
31621	0,603	0,910	0,605	0,706	16,9070	23,65	3,55	80,74
31671	0,502	0,501	0,502	0,502	10,6281	3,1438	5,644	12,5345
31681	0,385	0,700	0,380	0,488	10,6280	3,26	5,6441	13,28
316x1	0,71	0,848	0,709	0,756	16,9119	8,3432	3,5468	26,5528
316y1	0,441	0,610	0,438	0,496	10,6280	3,17	5,64	12,69
31711	0,807	0,799	0,809	0,805	10,8133	18,19	5,55	94,12
31721	0,592	0,931	0,591	0,705	10,8100	61,49	5,55	332,99
31771	0,504	0,499	0,504	0,502	6,7469	4,85	8,89	32,76
31781	0,370	0,712	0,369	0,484	6,7469	5,15	8,89	35,37
317x1	0,704	0,851	0,703	0,753	10,8137	20,52	5,55	107,19
317y1	0,423	0,624	0,425	0,491	6,7469	4,91	8,89	33,32
311111221	0,799	0,797	0,798	0,798	16,8633	68,07	3,55	234,23
311211221	0,764	0,765	0,762	0,764	15,5052	1277,98	3,87	4543,26
311711221	0,502	0,501	0,502	0,502	10,6248	12,61	5,64	55,18
311811221	0,501	0,503	0,503	0,502	10,6246	13,54	5,64	60,83
311x11221	0,794	0,797	0,799	0,797	16,8051	175,91	3,57	624,32
311y11221	0,501	0,502	0,507	0,503	10,6246	12,7518	5,6438	56,0703
312111221	0,807	0,799	0,809	0,805	10,7627	74,5153	5,5744	405,3373
312211221	0,815	0,813	0,815	0,814	7,9887	2007,1321	7,5154	11088,1
312711221	0,504	0,499	0,504	0,502	6,7453	11,9187	8,8923	86,417
312811221	0,525	0,530	0,520	0,525	6,7444	22,3164	8,896	176,8408
312x11221	0,778	0,782	0,779	0,780	10,0134	1103,6204	5,9898	6120,71
312y11221	0,510	0,506	0,509	0,508	6,7454	13,21	8,89	97,54
316111221	0,799	0,797	0,798	0,798	16,8633	68,07	3,55	234,23
316211221	0,805	0,806	0,807	0,806	15,8021	963,25	3,80	3418,91
316711221	0,502	0,501	0,502	0,502	10,6248	12,614	5,6436	55,1818
316811221	0,505	0,499	0,502	0,502	10,6246	13,30	5,64	59,33
316x11221	0,794	0,795	0,799	0,796	16,7436	78,6089	3,5825	271,6965
316y11221	0,500	0,503	0,499	0,501	10,6246	12,70	5,64	55,84
317111221	0,807	0,799	0,809	0,805	10,7627	74,5153	5,5744	405,3373
317211221	0,772	0,782	0,779	0,778	10,0547	1032,18	5,97	5717,85
317711221	0,504	0,499	0,504	0,502	6,7453	11,92	8,89	86,42
317811221	0,506	0,502	0,509	0,506	6,7456	12,76	8,89	93,61
317x11221	0,803	0,801	0,806	0,803	10,7348	101,24	5,59	555,60
317y11221	0,503	0,504	0,504	0,504	6,7454	12,08	8,89	87,96
311111222	0,792	0,796	0,792	0,793	16,6808	235,00	3,60	794,06
311211222	0,787	0,791	0,789	0,789	14,7438	1501,24	4,07	5264,06
311711222	0,502	0,501	0,502	0,502	10,5794	85,46	5,64	368,15
311811222	0,509	0,504	0,508	0,507	10,5766	88,43	5,64	384,82

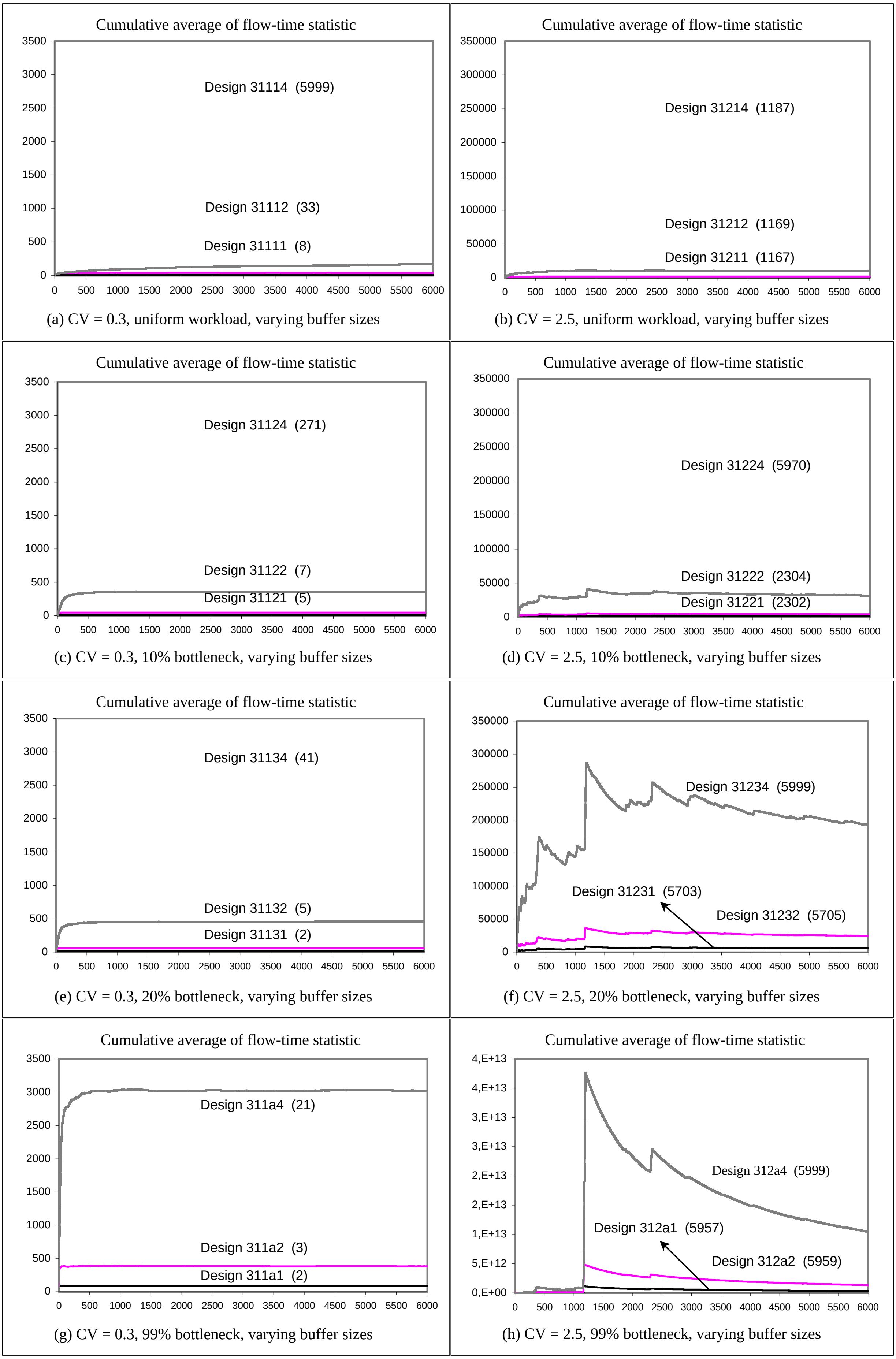
Table D.2 Summary statistics for job-shop experiments (continued)

Design	Utilization				Hourly		Interdeparture Time (min)	Time in System (min.)
	M/C #1	M/C #2	M/C #3	Average	Throughput	WIP		
311x11222	0,791	0,795	0,789	0,792	16,5179	721,46	3,63	2524,52
311y11222	0,502	0,499	0,506	0,502	10,5828	85,34	5,64	367,22
312111222	0,802	0,790	0,794	0,795	10,5908	349,81	5,66	1853,67
312211222	0,796	0,804	0,799	0,800	7,5622	2090,04	7,94	11880,30
312711222	0,503	0,497	0,502	0,501	6,7460	67,41	8,88	464,81
312811222	0,524	0,523	0,527	0,525	6,7363	97,45	8,89	745,58
312x11222	0,807	0,806	0,812	0,808	9,7981	1438,47	6,05	8963,62
312y11222	0,507	0,512	0,509	0,509	6,7464	69,85	8,88	486,84
316111222	0,792	0,796	0,792	0,793	16,6808	235,00	3,60	794,06
316211222	0,794	0,798	0,801	0,798	14,7518	1589,36	4,06	5915,53
316711222	0,502	0,501	0,502	0,502	10,5794	85,4587	5,6405	368,1502
316811222	0,505	0,506	0,504	0,505	10,5773	87,33	5,64	378,66
316x11222	0,787	0,791	0,790	0,789	16,5232	679,99	3,6294	2368,47
316y11222	0,500	0,502	0,501	0,501	10,5828	85,26	5,64	366,70
317111222	0,802	0,790	0,794	0,795	10,5908	349,81	5,66	1853,67
317211222	0,774	0,774	0,772	0,773	9,9581	1531,56	6,02	8601,10
317711222	0,503	0,497	0,502	0,501	6,7460	67,40	8,88	464,81
317811222	0,505	0,503	0,505	0,504	6,7463	70,16	8,88	489,58
317x11222	0,820	0,819	0,817	0,819	10,6859	401,96	5,61	2183,76
317y11222	0,503	0,502	0,501	0,502	6,7465	67,81	8,88	468,62

Table D.2 Summary statistics for job-shop experiments (continued)

Design	Utilization										Hourly		Interdeparture Time (min)	Time in System (min.)
	M/C #1	M/C #2	M/C #3	M/C #4	M/C #5	M/C #6	M/C #7	M/C #8	M/C #9	Average	Throughput	WIP		
91111	0,799	0,798	0,799	0,802	0,798	0,799	0,795	0,799	0,798	0,799	16,9045	20,89	3,55	70,72
91171	0,502	0,501	0,502	0,504	0,502	0,502	0,500	0,502	0,501	0,502	10,6264	7,78	5,64	38,43
91211	0,807	0,799	0,808	0,822	0,807	0,809	0,792	0,802	0,802	0,805	10,7984	53,70	5,55	297,28
91271	0,504	0,498	0,504	0,513	0,504	0,505	0,495	0,501	0,501	0,502	6,7446	12,39	8,89	99,98
911111221	0,800	0,797	0,798	0,796	0,797	0,799	0,797	0,799	0,797	0,798	16,7468	241,99	3,58	858,60
911711221	0,504	0,503	0,503	0,502	0,504	0,504	0,503	0,504	0,503	0,503	10,5943	35,62	5,66	185,24
912111221	0,815	0,805	0,811	0,798	0,818	0,811	0,802	0,808	0,810	0,810	10,7245	267,56	5,60	1485,35
912711221	0,509	0,502	0,505	0,498	0,510	0,506	0,500	0,504	0,505	0,505	6,7257	33,26	8,92	280,11
911111222	0,807	0,804	0,807	0,809	0,808	0,808	0,804	0,804	0,810	0,806	16,4788	1089,92	3,64	3891,57
911711222	0,504	0,502	0,504	0,505	0,504	0,500	0,502	0,502	0,505	0,503	10,5339	220,25	5,69	1084,62
912111222	0,793	0,783	0,781	0,779	0,790	0,787	0,779	0,787	0,784	0,786	10,2181	1371,81	5,87	7755,91
912711222	0,506	0,497	0,503	0,495	0,507	0,503	0,497	0,501	0,502	0,502	6,6568	180,89	9,01	1449,09

APPENDIX E. SAME SCALE FIGURES



(a) CV = 0.3, uniform workload, varying buffer sizes

(b) CV = 2.5, uniform workload, varying buffer sizes

(c) CV = 0.3, 10% bottleneck, varying buffer sizes

(d) CV = 2.5, 10% bottleneck, varying buffer sizes

(e) CV = 0.3, 20% bottleneck, varying buffer sizes

(f) CV = 2.5, 20% bottleneck, varying buffer sizes

(g) CV = 0.3, 99% bottleneck, varying buffer sizes

(h) CV = 2.5, 99% bottleneck, varying buffer sizes

Figure 4.1 Experimental results for 3-stage serial line containing all reliable machines with a total processing time of 3 minutes per job.

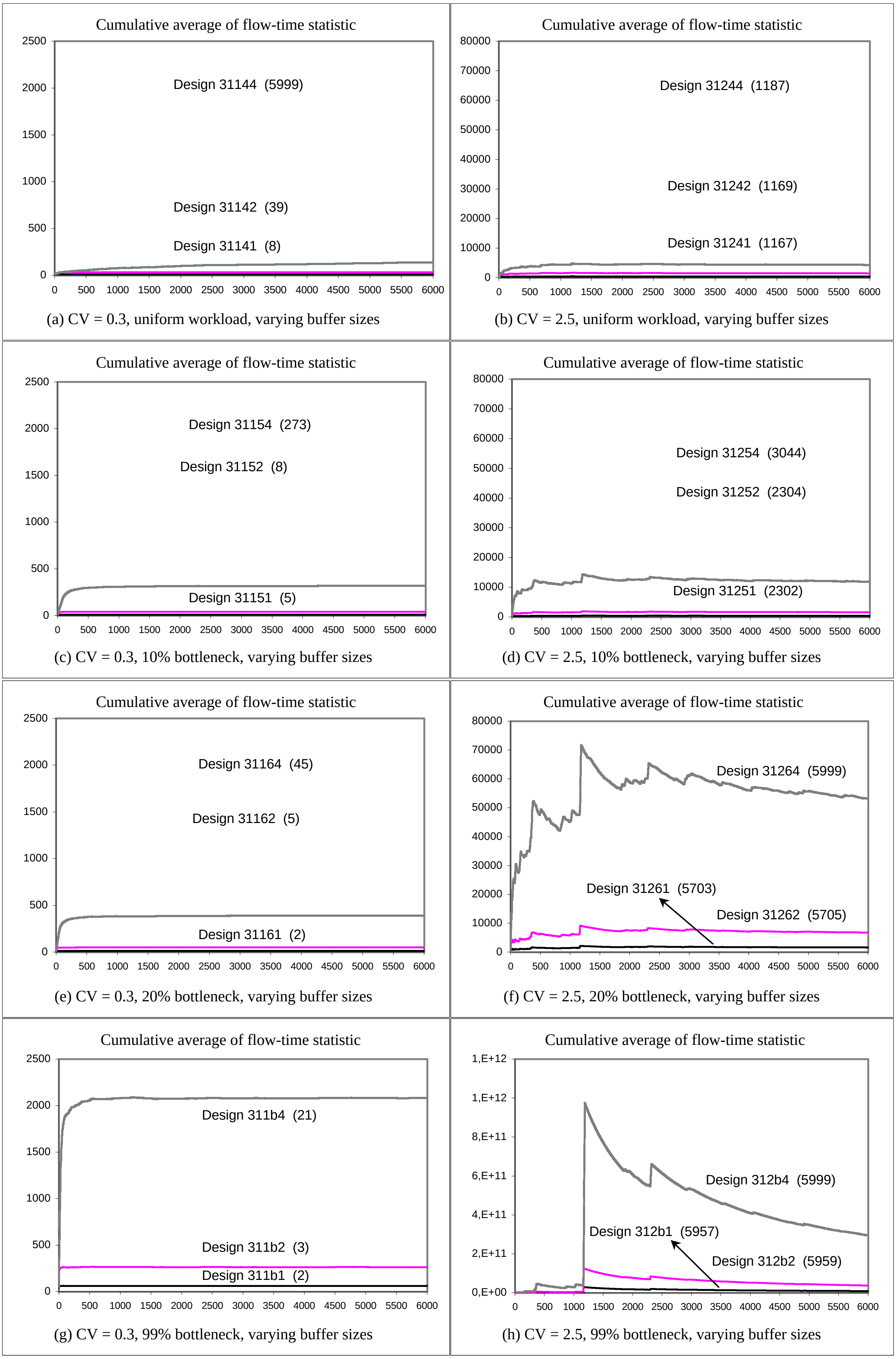


Figure 4.2 Experimental results for 3-stage serial line containing all reliable machines with a total processing time of 2.7 minutes per job.

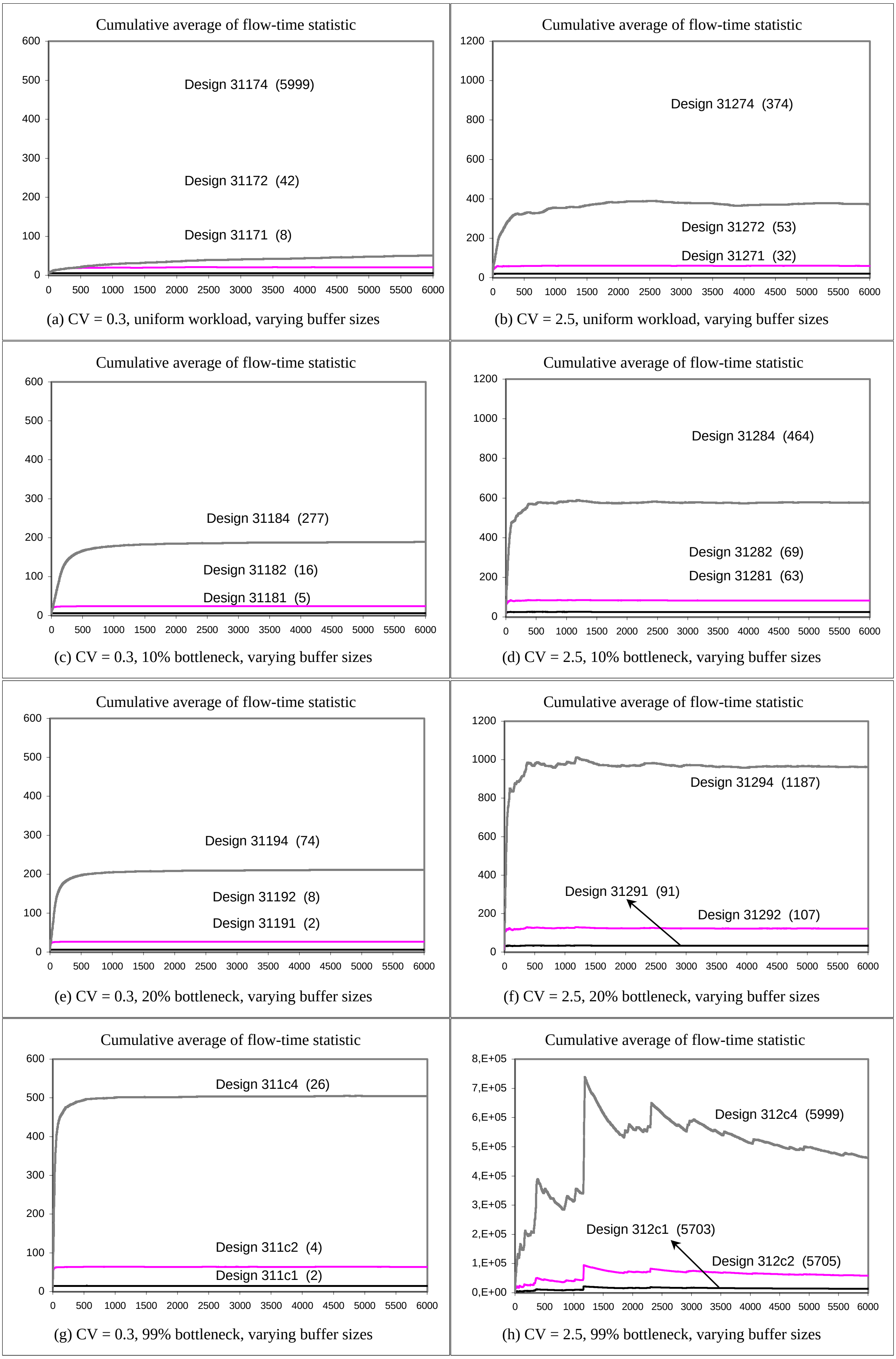
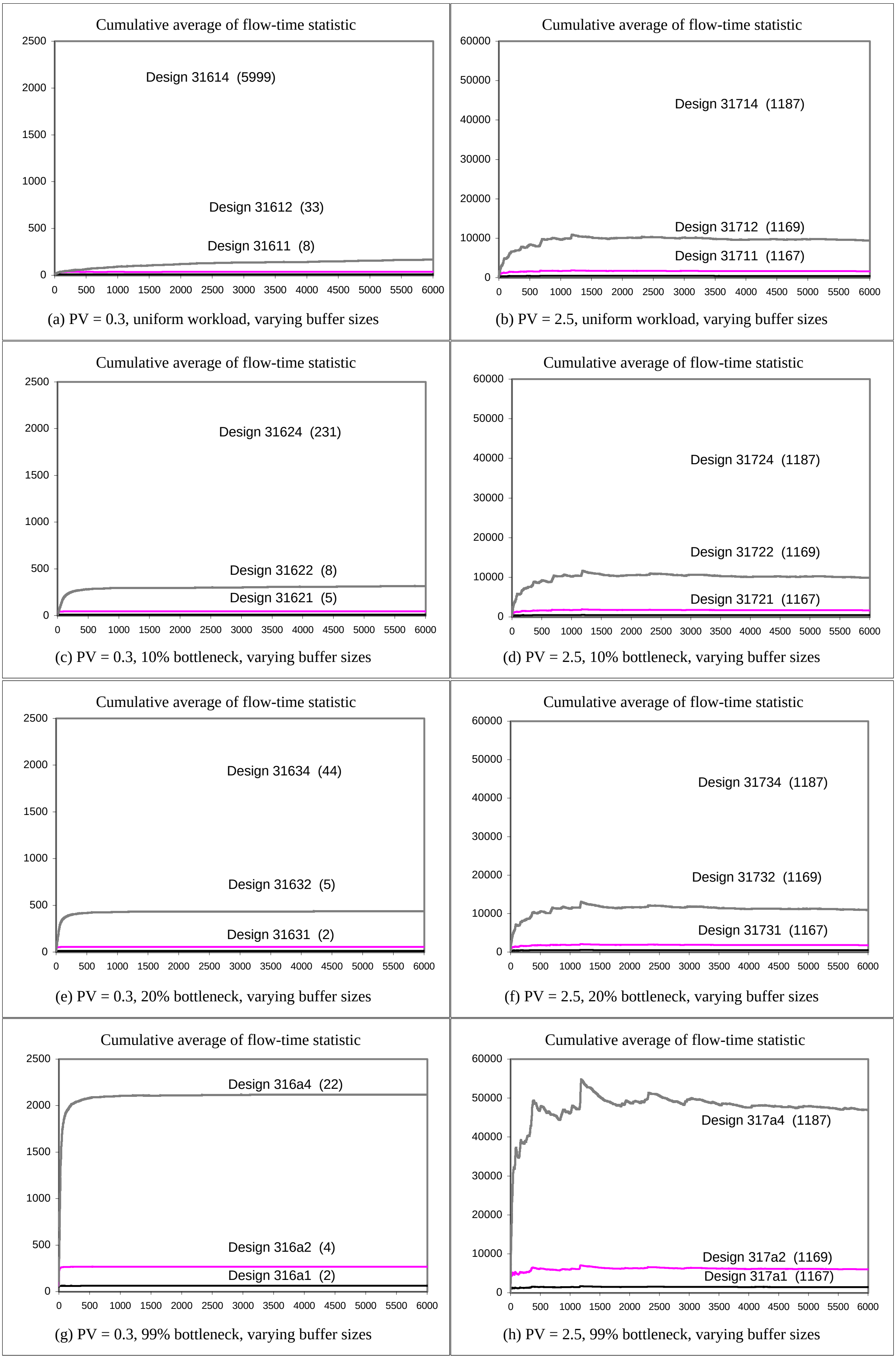


Figure 4.3 Experimental results for 3-stage serial line containing all reliable machines with a total processing time of 1.5 minutes per job.



(a) PV = 0.3, uniform workload, varying buffer sizes

(b) PV = 2.5, uniform workload, varying buffer sizes

(c) PV = 0.3, 10% bottleneck, varying buffer sizes

(d) PV = 2.5, 10% bottleneck, varying buffer sizes

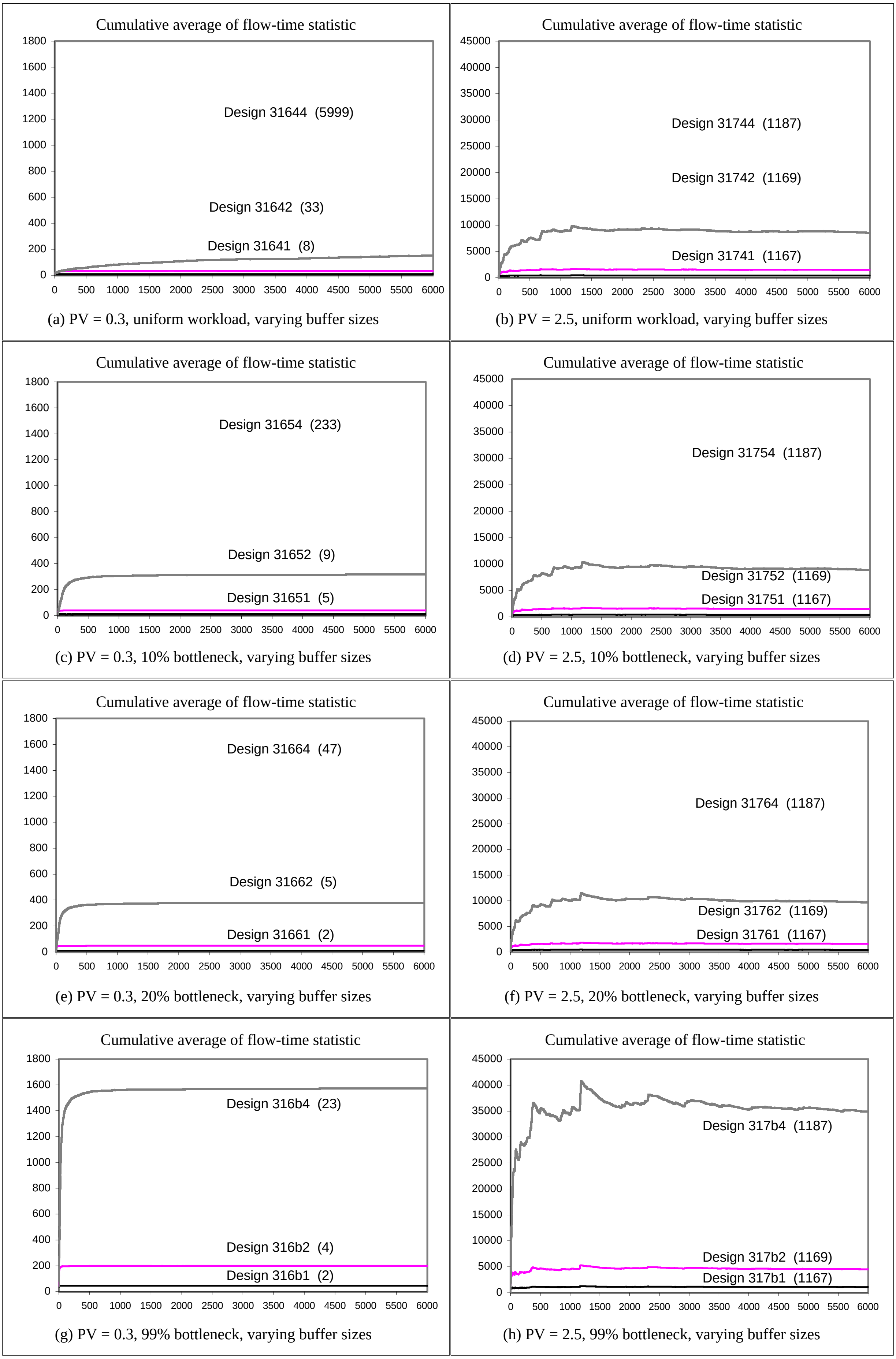
(e) PV = 0.3, 20% bottleneck, varying buffer sizes

(f) PV = 2.5, 20% bottleneck, varying buffer sizes

(g) PV = 0.3, 99% bottleneck, varying buffer sizes

(h) PV = 2.5, 99% bottleneck, varying buffer sizes

Figure 4.4 Experimental results for 3-stage serial line containing all reliable machines with a total processing time of 3 minutes per job.



(a) PV = 0.3, uniform workload, varying buffer sizes

(b) PV = 2.5, uniform workload, varying buffer sizes

(c) PV = 0.3, 10% bottleneck, varying buffer sizes

(d) PV = 2.5, 10% bottleneck, varying buffer sizes

(e) PV = 0.3, 20% bottleneck, varying buffer sizes

(f) PV = 2.5, 20% bottleneck, varying buffer sizes

(g) PV = 0.3, 99% bottleneck, varying buffer sizes

(h) PV = 2.5, 99% bottleneck, varying buffer sizes

Figure 4.5 Experimental results for 3-stage serial line containing all reliable machines with a total processing time of 2.7 minutes per job.

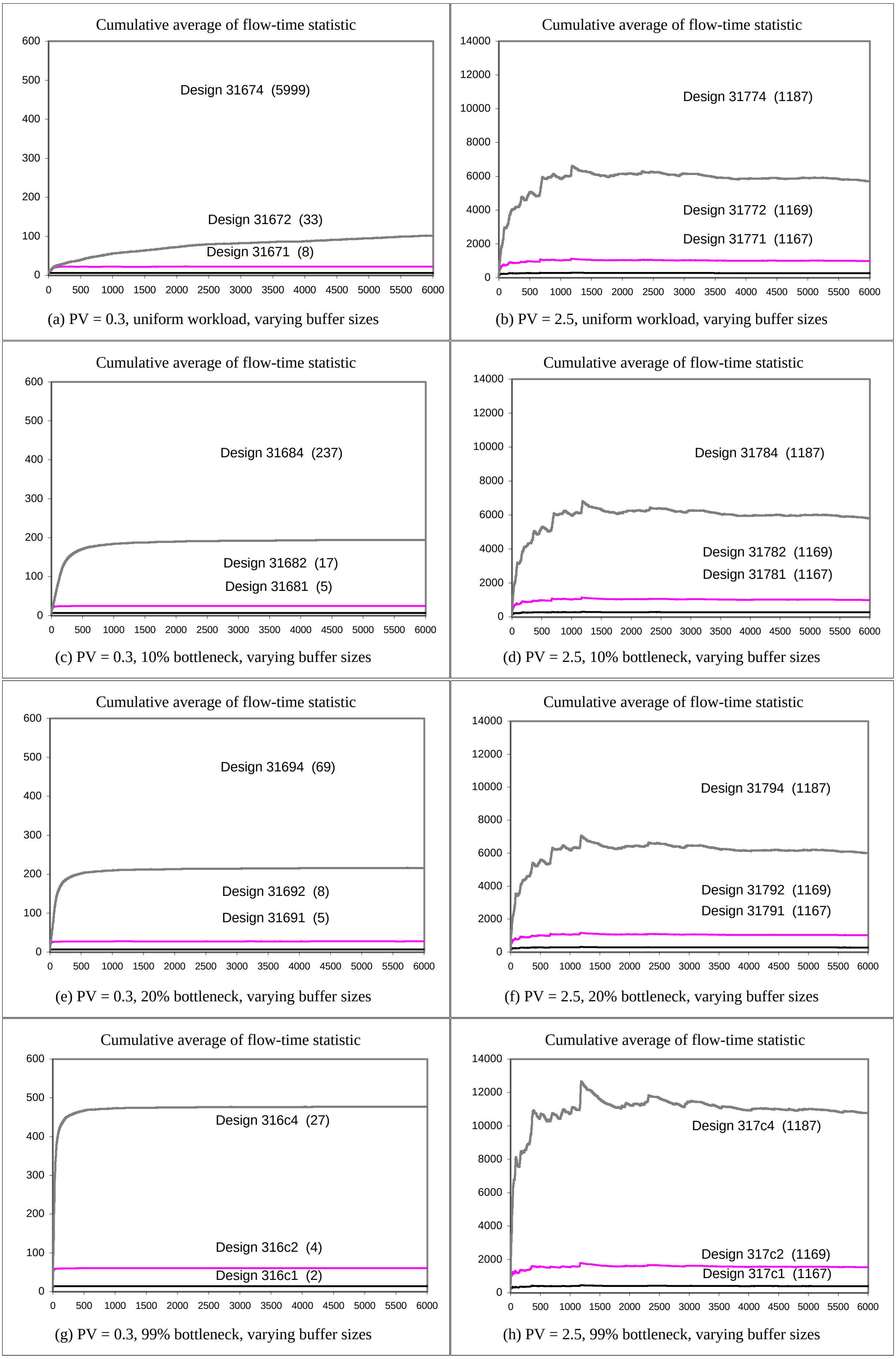
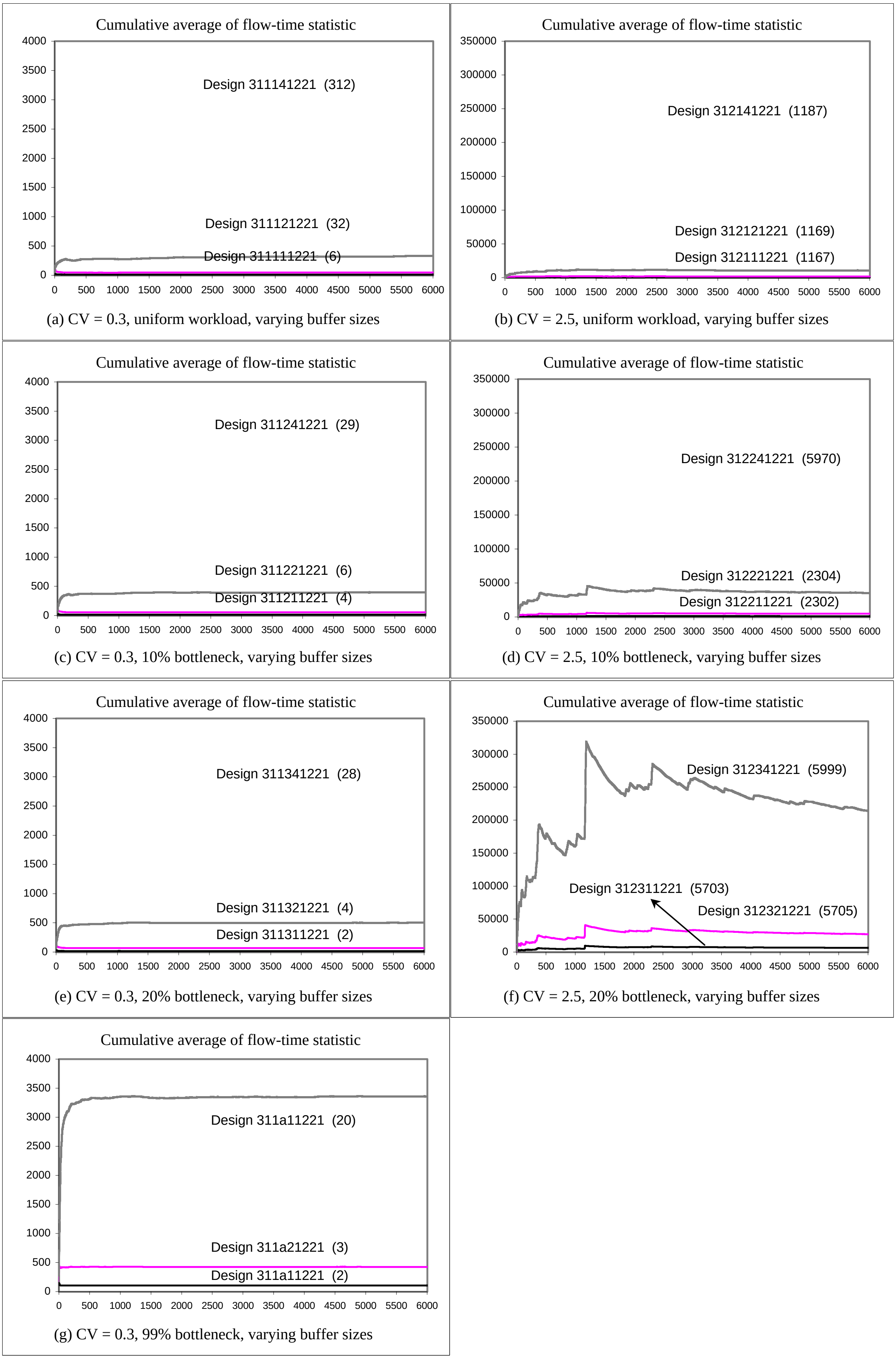


Figure 4.6 Experimental results for 3-stage serial line containing all reliable machines with a total processing time of 1.5 minutes per job.



(a) CV = 0.3, uniform workload, varying buffer sizes

(b) CV = 2.5, uniform workload, varying buffer sizes

(c) CV = 0.3, 10% bottleneck, varying buffer sizes

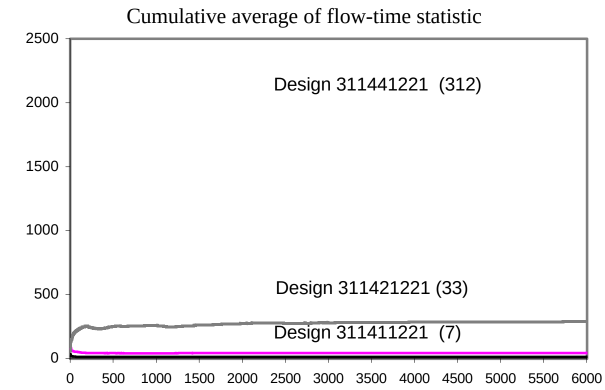
(d) CV = 2.5, 10% bottleneck, varying buffer sizes

(e) CV = 0.3, 20% bottleneck, varying buffer sizes

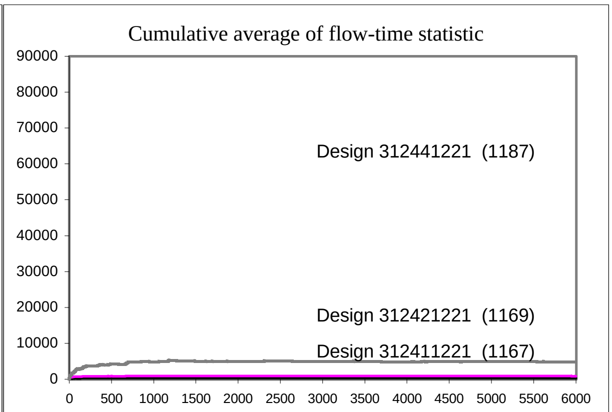
(f) CV = 2.5, 20% bottleneck, varying buffer sizes

(g) CV = 0.3, 99% bottleneck, varying buffer sizes

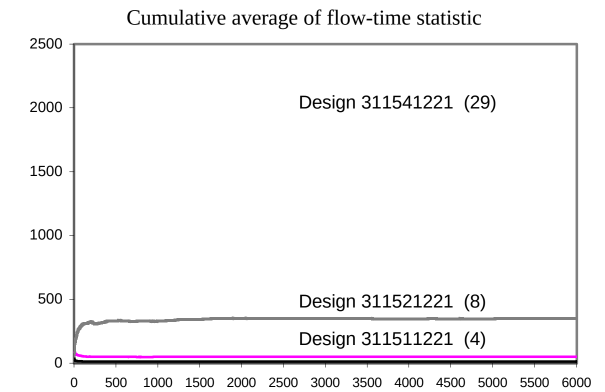
Figure 4.7 Experimental results for 3-stage serial line containing 90% unreliable machines with frequent breakdowns/short repair times and a total processing time of 3 minutes per job.



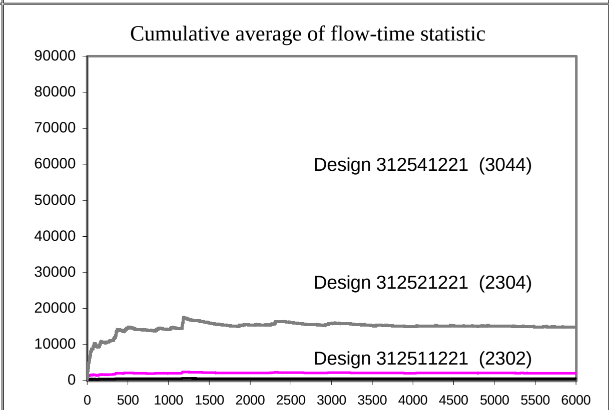
(a) CV = 0.3, uniform workload, varying buffer sizes



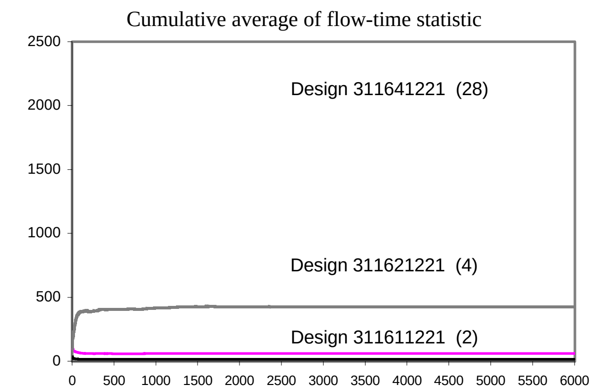
(b) CV = 2.5, uniform workload, varying buffer sizes



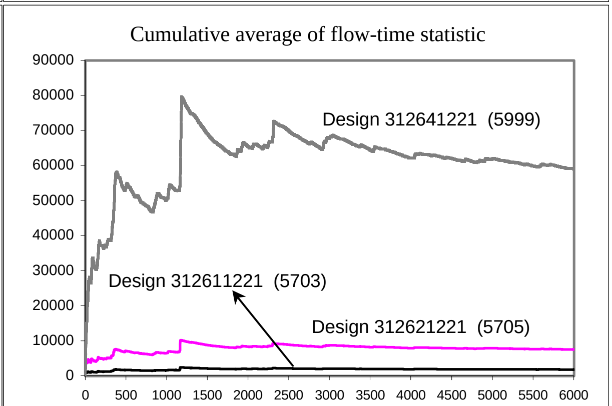
(c) CV = 0.3, 10% bottleneck, varying buffer sizes



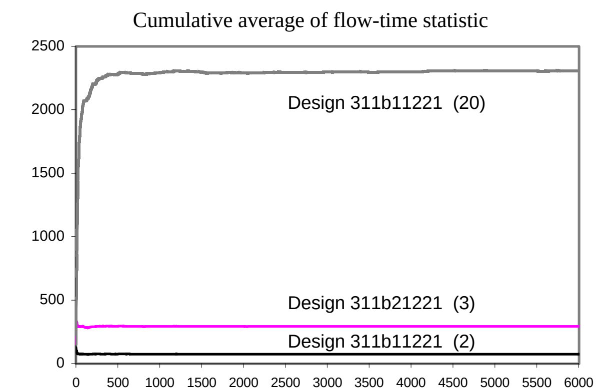
(d) CV = 2.5, 10% bottleneck, varying buffer sizes



(e) CV = 0.3, 20% bottleneck, varying buffer sizes



(f) CV = 2.5, 20% bottleneck, varying buffer sizes



(g) CV = 0.3, 99% bottleneck, varying buffer sizes

Figure 4.8 Experimental results for 3-stage serial line containing 90% unreliable machines with frequent breakdowns/short repair times and a total processing time of 2.7 minutes per job.

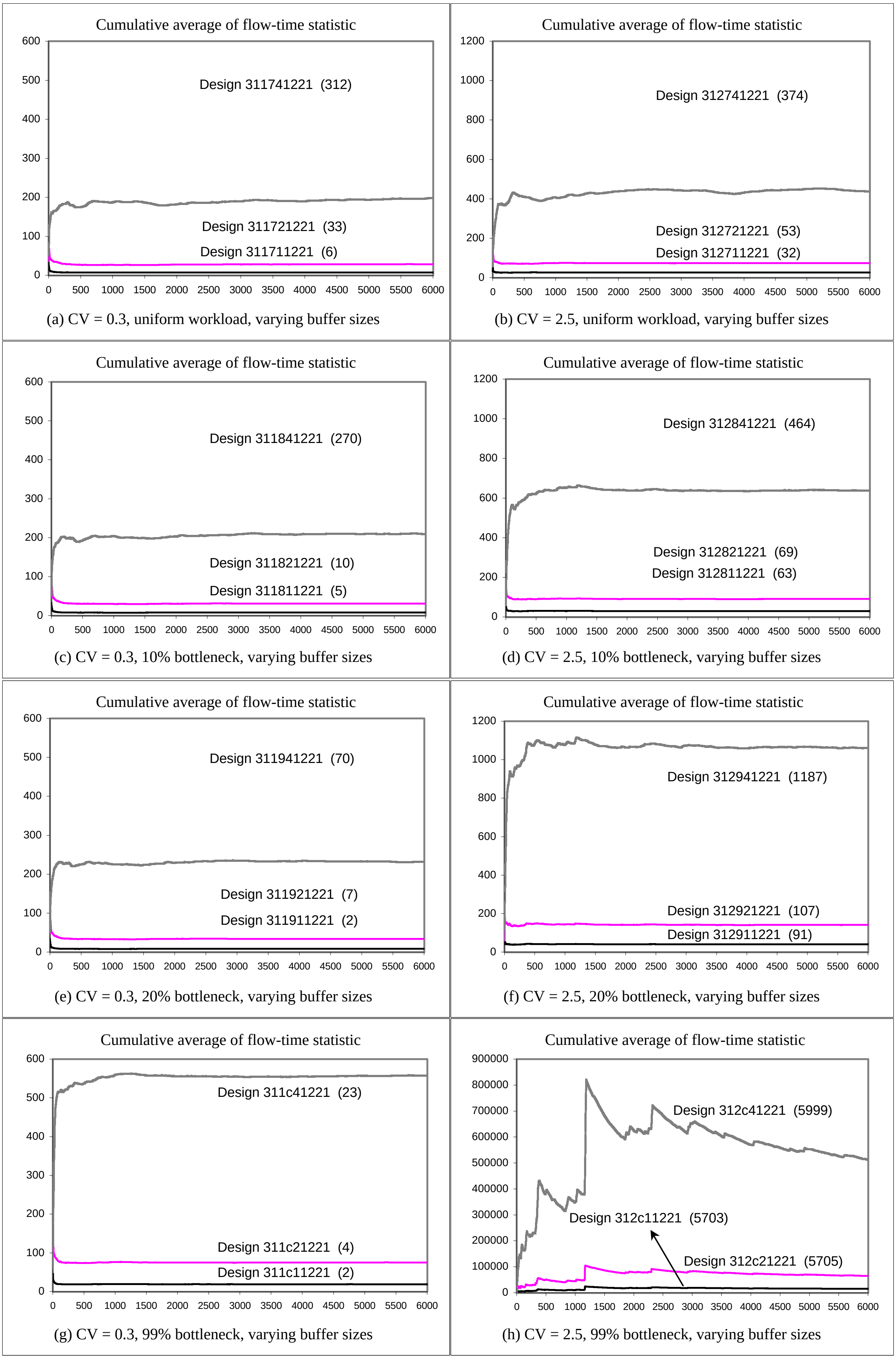
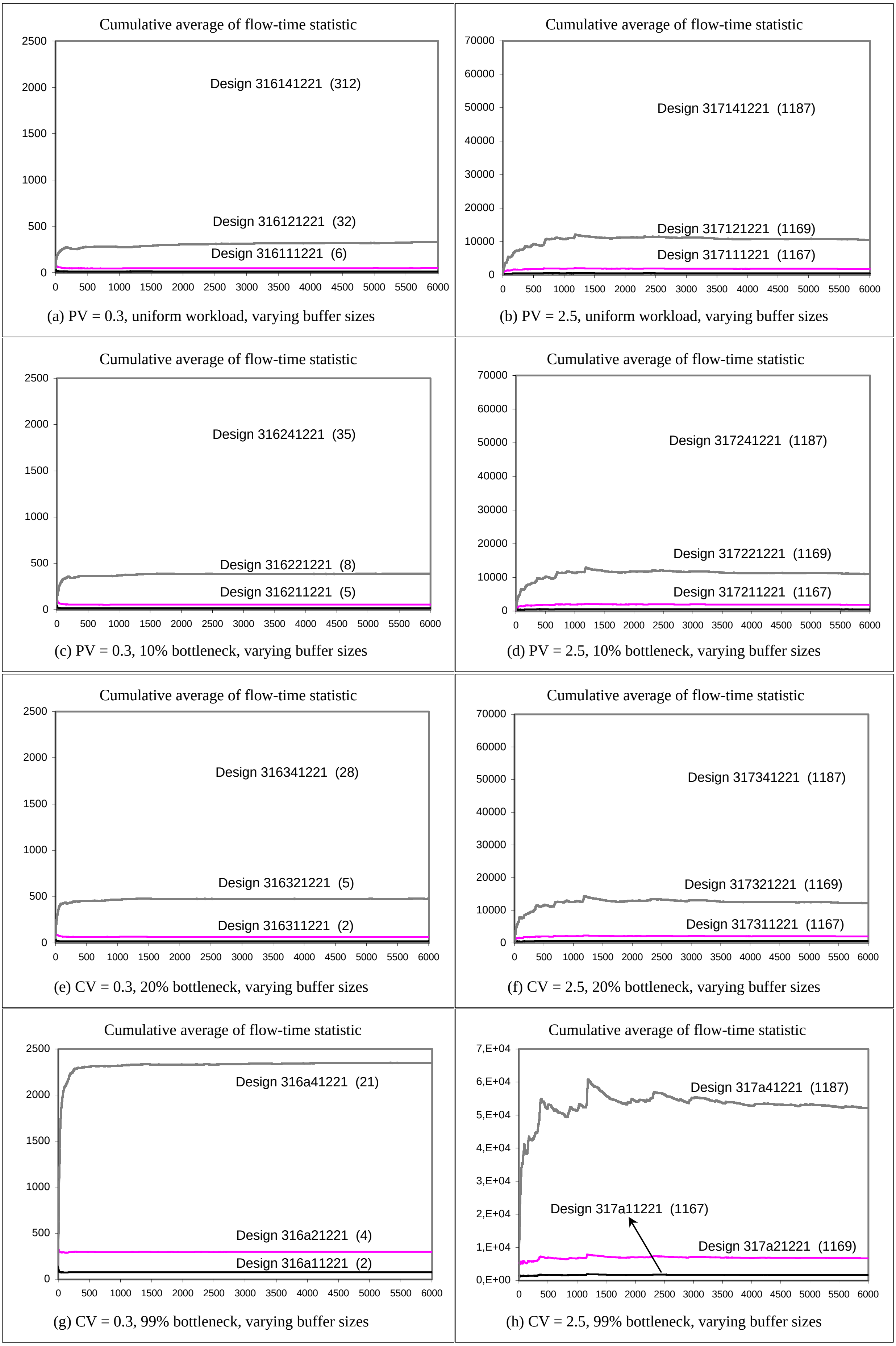


Figure 4.9 Experimental results for 3-stage serial line containing 90% unreliable machines with frequent breakdowns/short repair times and a total processing time of 1.5 minutes per job.



(a) PV = 0.3, uniform workload, varying buffer sizes

(b) PV = 2.5, uniform workload, varying buffer sizes

(c) PV = 0.3, 10% bottleneck, varying buffer sizes

(d) PV = 2.5, 10% bottleneck, varying buffer sizes

(e) CV = 0.3, 20% bottleneck, varying buffer sizes

(f) CV = 2.5, 20% bottleneck, varying buffer sizes

(g) CV = 0.3, 99% bottleneck, varying buffer sizes

(h) CV = 2.5, 99% bottleneck, varying buffer sizes

Figure 4.10 Experimental results for 3-stage serial line containing 90% unreliable machines with frequent breakdowns/short repair times and a total processing time of 3 minutes per job.

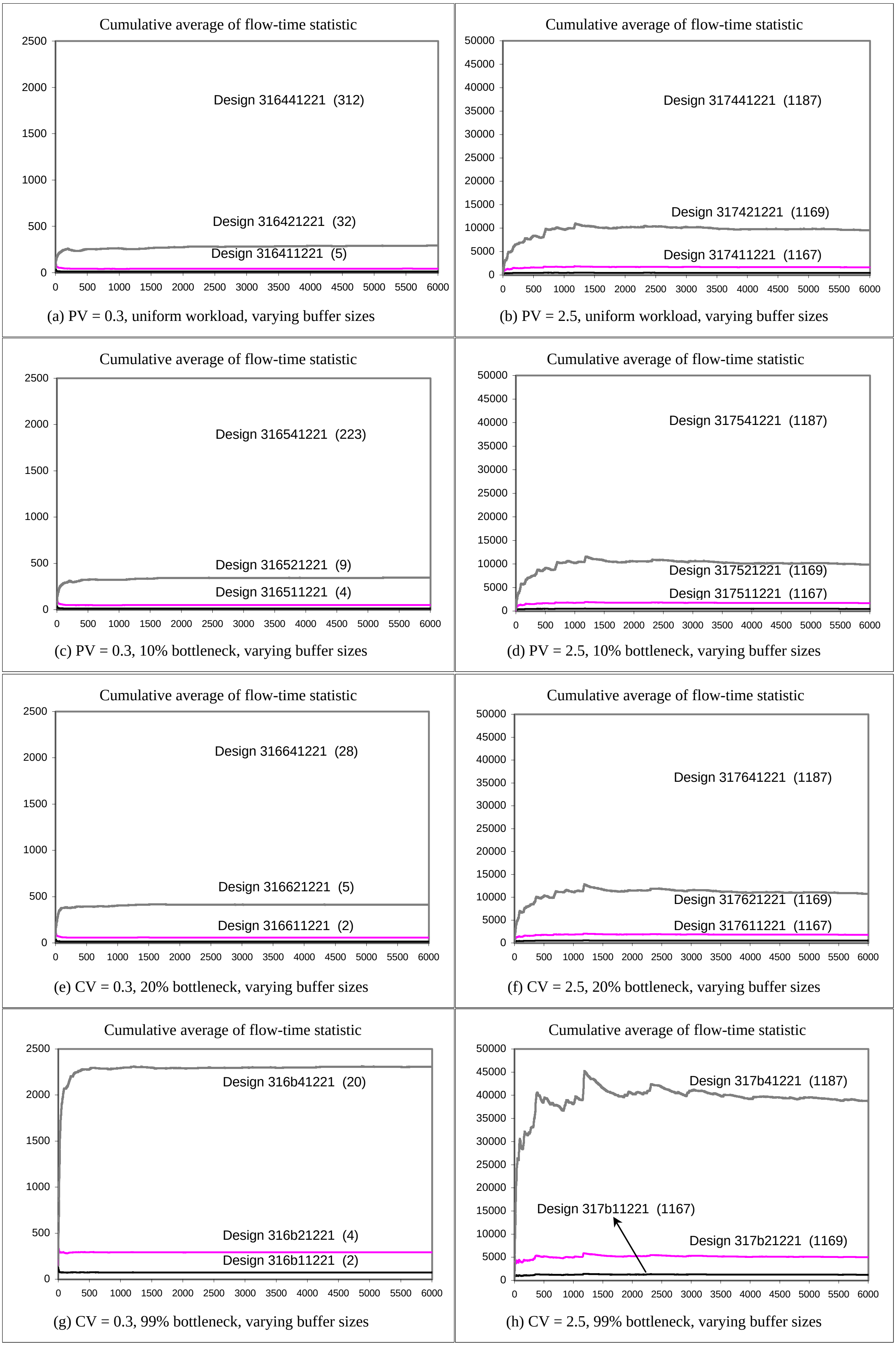
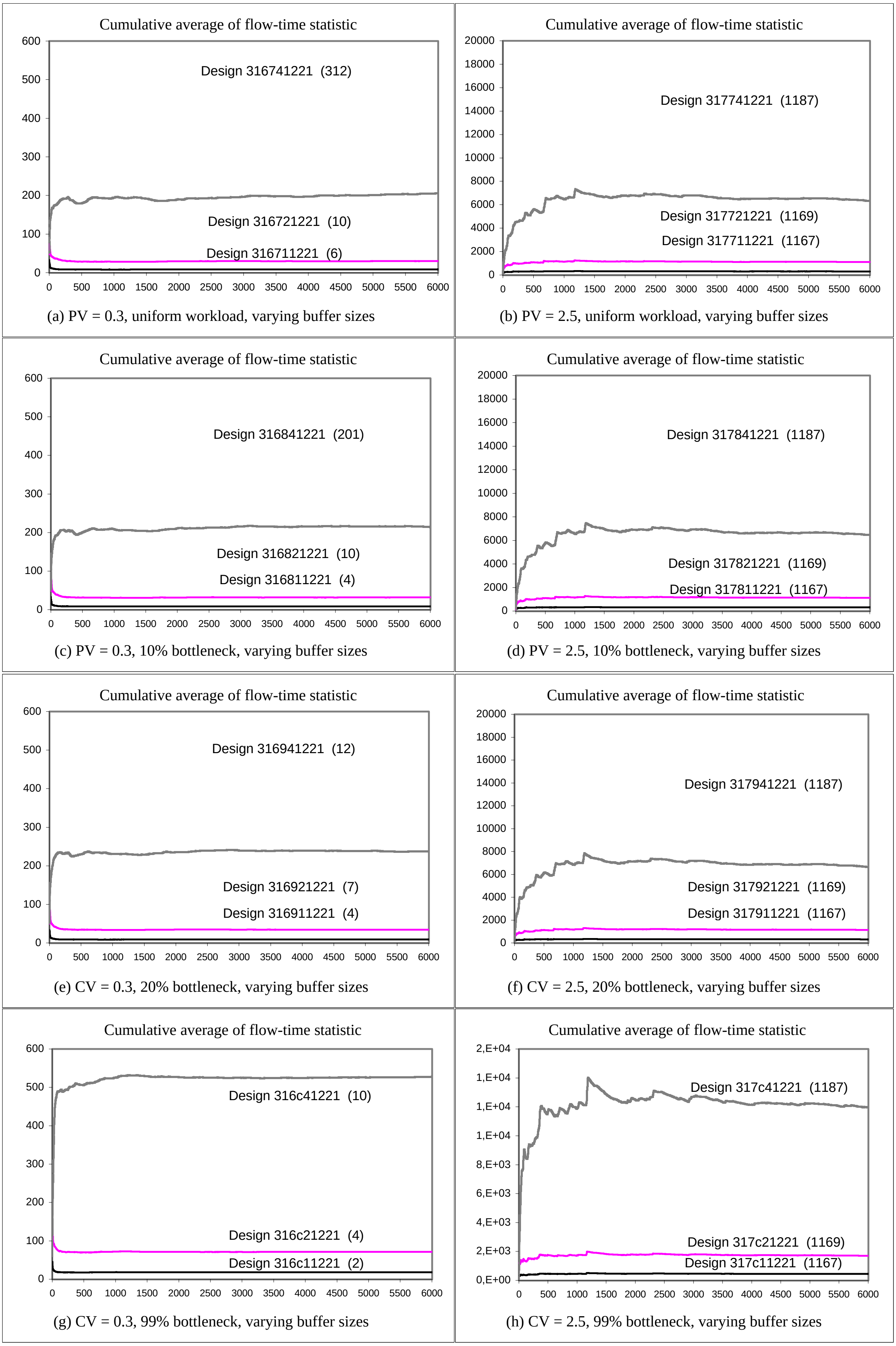


Figure 4.11 Experimental results for 3-stage serial line containing 90% unreliable machines with frequent breakdowns/short repair times and a total processing time of 2.7 minutes per job.



(a) PV = 0.3, uniform workload, varying buffer sizes

(b) PV = 2.5, uniform workload, varying buffer sizes

(c) PV = 0.3, 10% bottleneck, varying buffer sizes

(d) PV = 2.5, 10% bottleneck, varying buffer sizes

(e) CV = 0.3, 20% bottleneck, varying buffer sizes

(f) CV = 2.5, 20% bottleneck, varying buffer sizes

(g) CV = 0.3, 99% bottleneck, varying buffer sizes

(h) CV = 2.5, 99% bottleneck, varying buffer sizes

Figure 4.12 Experimental results for 3-stage serial line containing 90% unreliable machines with frequent breakdowns/short repair times and a total processing time of 1.5 minutes per job.

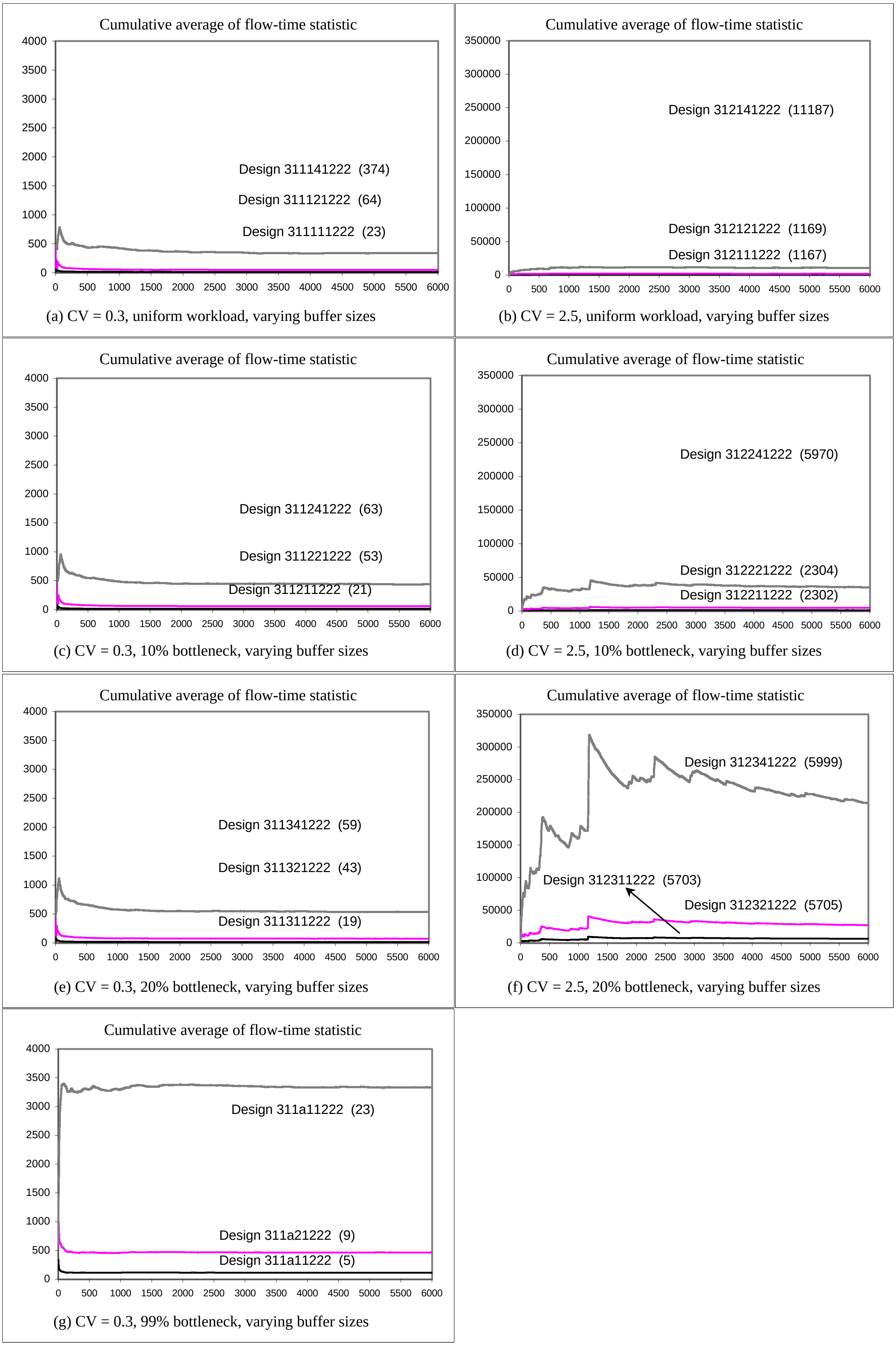
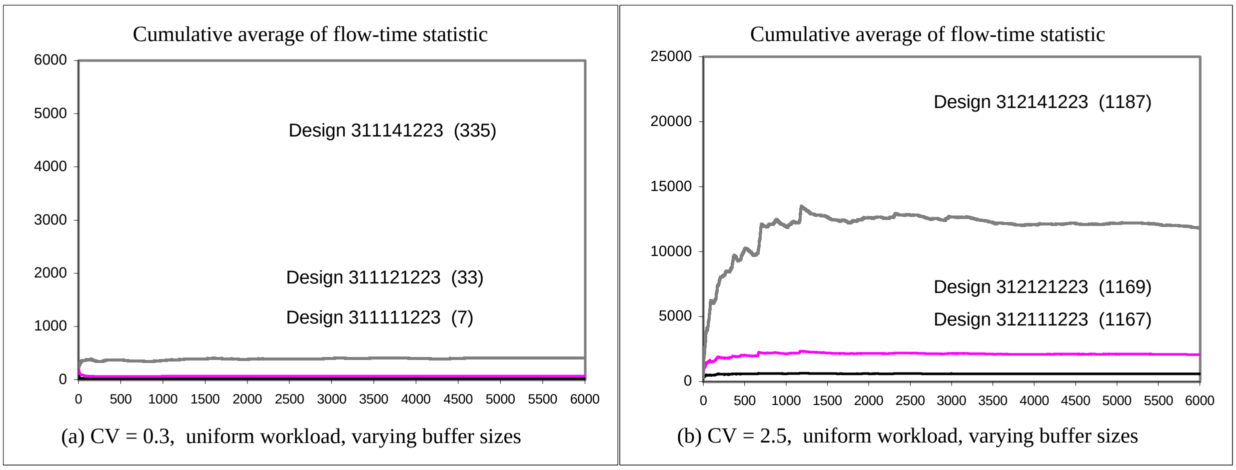


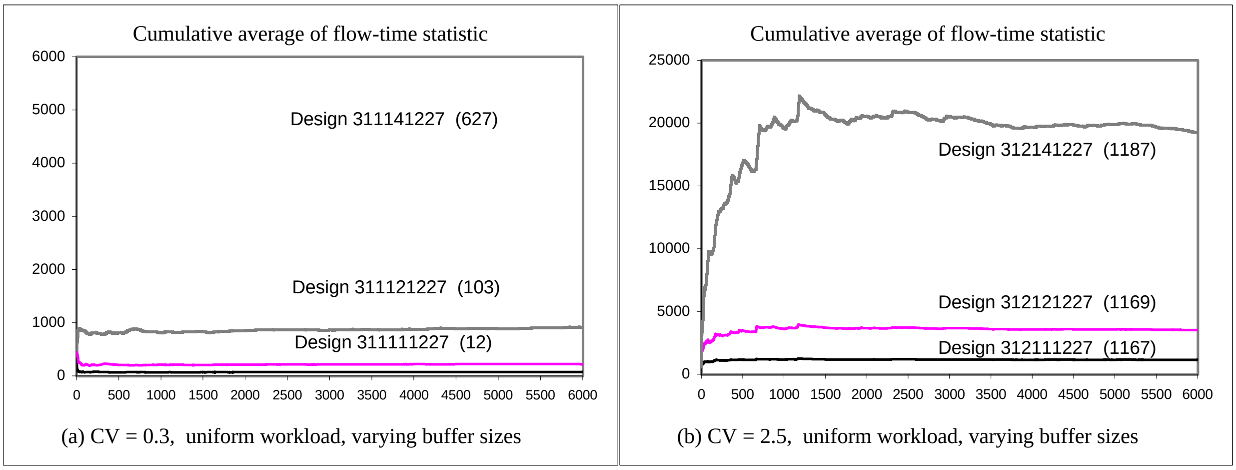
Figure 4.13 Experimental results for 3-stage serial line containing 90% unreliable machines with rare breakdowns/long repair times and a total processing time of 3 minutes per job.



(a) CV = 0.3, uniform workload, varying buffer sizes

(b) CV = 2.5, uniform workload, varying buffer sizes

Figure 4.14 Experimental results for 3-stage serial line containing 80% unreliable machines with frequent breakdowns/short repair times and a total processing time of 3 minutes per job.



(a) CV = 0.3, uniform workload, varying buffer sizes

(b) CV = 2.5, uniform workload, varying buffer sizes

Figure 4.15 Experimental results for 3-stage serial line containing 50% unreliable machines with frequent breakdowns/short repair times and a total processing time of 3 minutes per job.

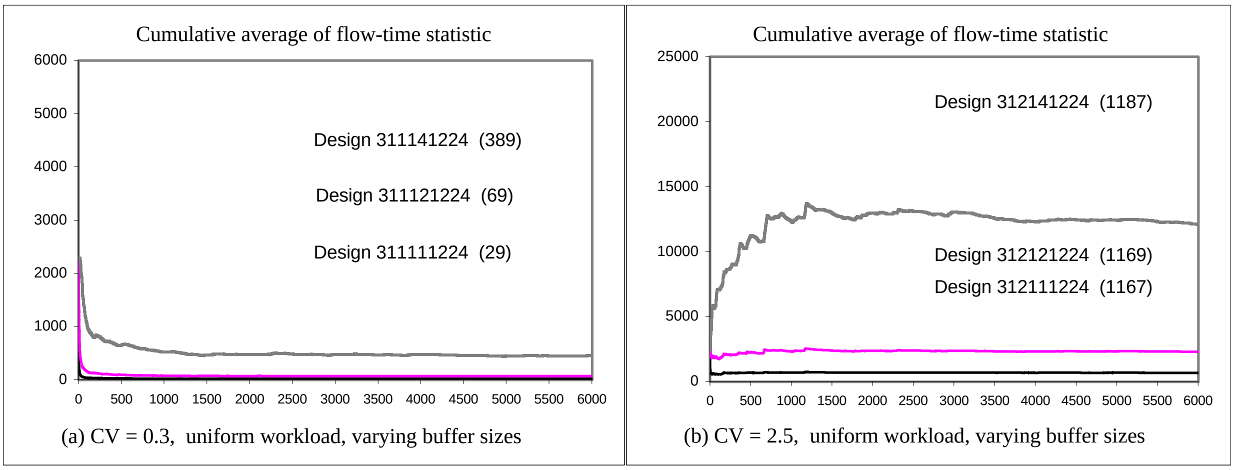


Figure 4.16 Experimental results for 3-stage serial line containing 80% unreliable machines with rare breakdowns/long repair times and a total processing time of 3 minutes per job.

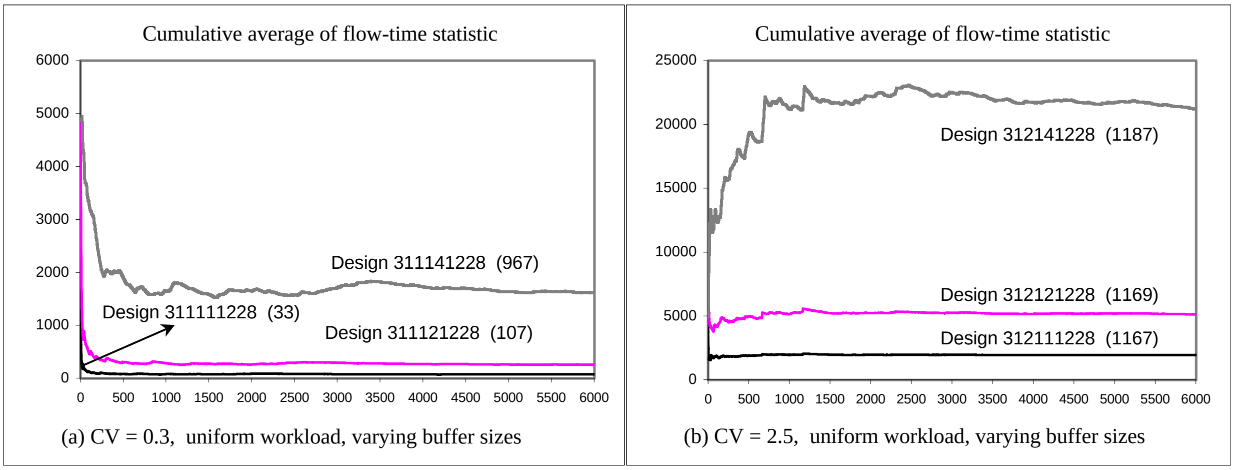
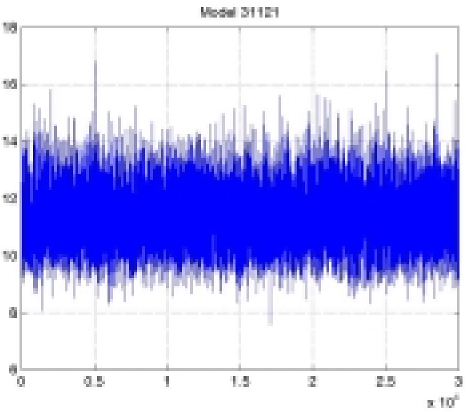
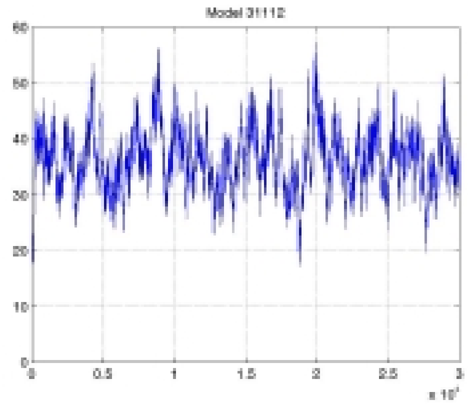


Figure 4.17 Experimental results for 3-stage serial line containing 50% unreliable machines with rare breakdowns/long repair times and a total processing time of 3 minutes per job.

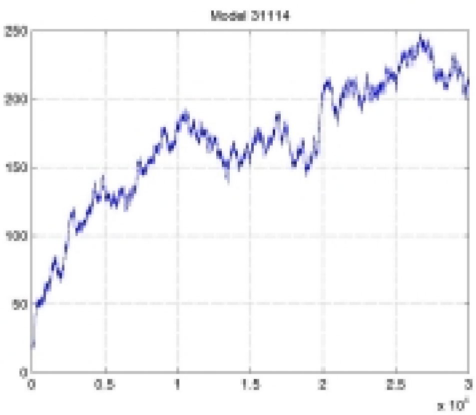
APPENDIX F. SAMPLE INDIVIDUAL PLOTS



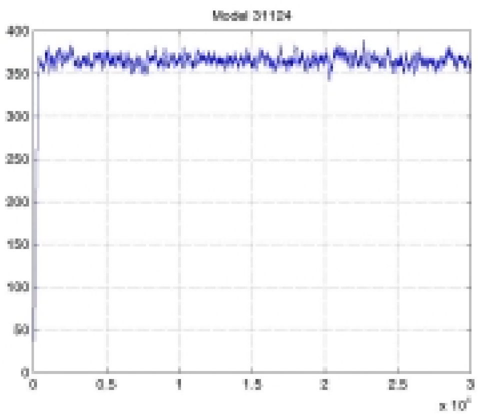
(a) Normal behavior



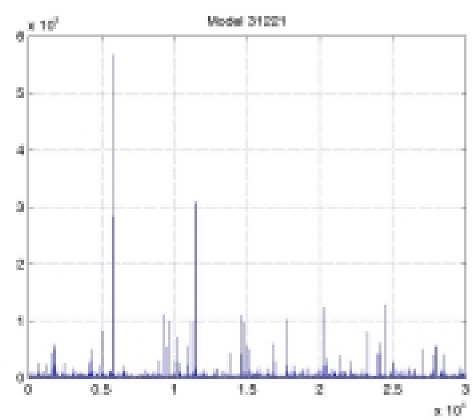
(b) Cyclic behavior



(c) Increasing



(d) Transient period



(e) Outliers

Figure F.1 Individual plots of flow-time statistic that does not help determining the transient period.

VITA

Burhaneddin Sandıkçı was born on June 16, 1978, in Samsun, Turkey. He received his high school education at Samsun Anadolu Lisesi. He attended to the Department of Industrial Engineering at Marmara University, İstanbul, Turkey, in 1995 and graduated with honors in July, 1999. He worked with Prof. M. Akif Eyler and Assoc. Prof Mustafa Özbayrak for his graduation thesis. In September 1999, he joined to the Department of Industrial Engineering at Bilkent University, Ankara, Turkey as a research assistant. From that time to present, he worked with Assoc Prof. İhsan Sabuncuoğlu for his graduate study. His research interests are applied operations research, simulation modeling and analysis, statistical analysis of data, and computer implementation of operational research techniques.