ANALYSIS OF VARIANCE REDUCTION TECHNIQUES IN VARIOUS SYSTEMS

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING AND THE INSTITUTE OF ENGINEERING AND SCIENCE OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

Sabri ÇELİK

January, 2003

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 for the degree of Master of Science.

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 for the degree of Master of Science.

Assist. Prof. Mehmet Murat Fadıloğlu (Co-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 for the degree of Master of Science.

Prof. Ülkü Gürler

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 for the degree of Master of Science.

Assist. Prof. Yavuz Günalay

Approved for the Institute of Engineering and Science:

Prof. Mehmet Baray
Director of Institute of Engineering and Science

ABSTRACT

ANALYSIS OF VARIANCE REDUCTION TECHNIQUES IN VARIOUS SYSTEMS

Sabri ÇELİK
M.S. in Industrial Engineering
Advisors: Professor İhsan SABUNCUOĞLU
Assistant Professor M. Murat FADILOĞLU
August, 2002

In this thesis, we consider four different Variance Reduction Techniques (VRTs): Antithetic Variates (AV), Latin Hypercube Sampling (LHS), Control Variates (CV), and Poststratified Sampling (PS). These methods individually or in combination are applied to the steady state simulation of three well-studied systems. These systems are M/M/1 Queuing System, a Serial Line Production System, and an (s,S) Inventory Policy. Our results indicate that there is no guarantee of a reduction in variance or an improvement in precision in estimates. The performance of VRTs totally depends on the system characteristics. Nevertheless, CV performs better than PS, AV and LHS on the average. Therefore, instead of altering the input part of the simulation, extracting more information by CV should be more effective. However, if any extra information about the system is not available, AV or LHS can be favored since they do not require additional knowledge about the system. Furthermore, since the analysis of output data through CV or PS requires a negligible time compared to the simulation run time, applying CV and PS at all possible cases and then selecting the best one can be the best strategy in the variance reduction. The use of the combination of methods provides more improvement on the average.

Keywords: Simulation, Variance Reduction Techniques, Output Data Analysis, Antithetic Variates, Latin Hypercube Sampling, Control Variates, Poststratified Sampling.

ÖZET

VARYANS AZALTMA TEKNİKLERİNİN ÇEŞİTLİ SİSTEMLER ÜZERİNDE ANALİZİ

Sabri ÇELİK
Endüstri Mühendisliği, Yüksek Lisans
Danışmanlar: Prof. Dr. İhsan SABUNCUOĞLU
Yrd. Doç. Dr. M. Murat FADILOĞLU
Ağustos, 2002

Bu çalışmada dört farklı varyans azaltma tekniği ele alınmıştır: Antithetic Variates (AV), Latin Hypercube Sampling (LHS), Control Variates (CV), and Poststratified Sampling (PS). Bu teknikler tekli ve ikili gruplar halinde üç değişik sisteme uygulanmış ve tekniklerin sistemler durağan duruma ulaştıktan sonraki performansları incelenmiştir. Bu sistemler ise M/M/1 kuyruk sistemi, seri üretim hattı, ve (s,S) Envanter modelinden oluşmaktadır. Sonuçlara göre, varyansta bir azalma ya da güven aralığında bir gelişme garanti edilememektedir. Tekniklerin tekli ya da ikili performanslari tamamen sistemin yapısına ve özelliklerin bağlı bulunmaktadır. Bununla birlikte CV'nin PS, AV ve LHS'den daha iyi performans gösterdiği gözlenmiştir. Dolayısıyla sistemin verileriyle ilgilenmek yerine sistemin çıktılarından CV aracılığıyla daha fazla bilgi çıkarmaya çalışmak daha yararlıdır. Ama eğer bu işlem çok zor ise AV ya da LHS'de tercih edilebilir. Ayrıca CV ve PS'nin sistemin çıktılarını analiz etmesi gereken zaman benzetimin bilgisayar ortamında çalışmasıyla geçirilen zaman yanında ihmal edilebileceği için CV ve PS'yi mümkün olan bütün alternatifleriyle uygulayıp bunlar arasından en iyisini seçmek varyans azaltma konusunda en iyi strateji olabilmektedir. Son olarak tekniklerin ikili uygulamaları genel olarak tekli uygulamalardan daha iyi sonuç vermektedir.

Anahtar Sözcükler: Benzetim, Varyans Azaltma Teknikleri, Veri Çıktı Analizi, Antithetic Variates, Latin Hypercube Sampling, Control Variates, Poststratified Sampling.

Dedicated to

My Wonderful Parents

Zeynep & Mehmet ÇELİK

and

My Wonderful Brothers

Erkan & Ersin ÇELİK

ACKNOWLEDGEMENTS

I am deeply indebted to Dr. İhsan Sabuncuoğlu and I would like to express my deepest gratitude for his supervision, encouragement, and never-ending morale support throughout all stages of my studies. He devoted so much time and effort to the completion of my studies and without him I would never have finished this thesis. With my sincerity, I can only aspire to be like him if I ever become an academician or researcher in the future.

I am also indebted to Dr. Murat Fadıloğlu who also supervised me throughout all stages of my studies. I would like to present my special thanks to him for his supervision, encouragement and contribution to my thesis. He contributed much to this thesis with his creative and skeptical approach to the subject.

I would like to thank to Ülkü Gürler and Yavuz Günalay for accepting to read and review this thesis. I am very grateful to them for their tolerance and assistance.

Special thanks to Banu Yüksel for her invaluable assistance and friendship not only during this research but also throughout all my graduate studies. Her nice personality and knowledge encouraged me to do the best all the time.

I am also grateful to my colleagues Rabia Kayan, Savaş Çevik, Halil Şekerci, Batuhan Kızılışık, Aykut Özsoy and my friends Aytül Çatal, Gökçe Özcan, Özge Şahin, Sercan Özbay, Nurşen Töre, Utku Kumru, Yakup Bayram, and Utku Barın for their friendship and excellent morale support and encouragement all the time.

Finally, my special thanks and gratitude are for my wonderful family, the only reality in my life. With their unlimited love and support throughout all my life, I feel full self-confidence and believe that I will overcome all the obstacles in my life and do everything I want and only for my dearest parents and brothers.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Review

1.1. Introduction

Analytical methods may not be available or difficult to apply all the time in order to analyze the complex systems including stochastic input variables. In that case, numerical methods are recommended in order to analyze or at least to get an idea about the system behavior or performance. In this context, simulation has a widespread usage with the increasing availability and the capability of the computers. Thus currently it is one of the mostly used tools in practical cases. Simulation is the process of designing a model of the real system and conducting experiments with this model for the purpose of understanding the behaviour of the system and/or evaluating various strategies for the operation of the system (Pegden, Shannon and Sadowski [27])

Since the random inputs to any simulation model produce random outputs, further statistical analysis is required to better interpret the results of the simulation. In this manner, simulation just gives an estimate for each output variable and this estimate is calculated with some error depending on the purpose of the study and the desired precision. One way to increase the precision of the estimators is to increase the sample size. However, this may require a large amount of computational time and effort. Another way is the usage of some methods that provide the same precision with less simulation runs or result in more precision with the same number of simulation runs. Even though it may be sometimes very costly to reach satisfactory results, various techniques have been developed in order to decrease the anticipated cost.

Variance Reduction Techniques (VRTs) are experimental design and analysis techniques used to increase the precision of estimators without increasing the computational effort or to get the desired precision with less effort. A more statistical definition says that they are the techniques defining a new estimator, which has the same expectation as the default estimator but has a lower variance. That is, these techniques create a new unbiased estimator expected to have a lower variance than the default estimator. Biased estimators can be used in some cases as well (Schmeiser [31]). In those cases, the aim is to obtain a reduction in the mean square error (MSE).

Many VRTs have been developed since the beginning of the computers for the purpose of simulating a system. However, these methods require a thorough understanding of the model being simulated, or at least understanding of some relationships existing between the input and output random variables. In addition, as we will show throughout our experiments, the amount in the variance reduction cannot be predicted in advance and sometimes they can backfire and thus even a higher variance is obtained. Thereby some pilot runs of the simulation can be very useful not only to understand the relationships between the input and output random variables but also to guess the possible reduction in the variance that may be achieved by the application of the VRT.

Many studies focusing on the new methods or the classification of the methods have been made in the literature. Comprehensive surveys are available on the VRTs. A recent survey by L'Ecuyer [16] includes an overview of the techniques, efficiencies of those techniques and the related examples in the literature.

In this thesis, we will focus on four methods and their combinations. Namely, these are Antithetic Variates (AV), Latin Hypercube Sampling (LHS), Control Variates (CV) and Poststratified Sampling (PS). These methods either individually or in combination will be applied to the simulation results of three systems. Firstly, M/M/1 as the most basic queuing system is taken into account under two different utilization levels. A high utilization level (0.9) is chosen to examine the performances of the four methods and their combinations under highly utilized or variable systems. After that, the same process is performed for a moderately variable system and a lower utilization rate (0.5) is considered. Second, a well-studied system, a serial production system of five stations is taken into consideration. During this study, we assume that all stations have the same service time distribution and allow buffers of three units between the stations. Due to the limited (or finite) buffer spaces, 'starvation' and 'blockage' of the stations occur in this case as well. Finally, a simple inventory model, a periodic review inventory system controlled by an (s, S) Policy, is selected in order to analyze the behaviour of the four methods and their combinations. In this case, stochastic weekly demands exist and when the inventory position falls below the reorder point s, a new order is given with a constant lead time of two weeks.

All of the experiments are conducted using the SIMAN V program in the UNIX environment. The computer program codes for all cases of three systems are given in Appendix A. In these codes, only one model and experimental file are given for each case, i.e. only when Antithetic Variates is applied to interarrival and service times simultaneously.

This thesis is organized as follows. In the next section, we give an overview of the techniques in the literature, however, mostly focus on the relevant literature regarding the four techniques and their combinations. In Chapter 2, we present the experimental results of the single application of the techniques and the integration of the techniques on M/M/1 system and this is followed by the corresponding results of Serial Line Production System and (s, S) Inventory Policy in Chapter 3 and 4, respectively. After giving the detailed results, we discuss the general results at two points throughout each chapter. Finally, this thesis ends with a conclusion and further research directions in Chapter 6. In this chapter, we summarize the main findings and conclusions and recommend directions for further research.

1.2. Literature Review

Available literature on VRTs can be classified into three categories as shown below:

i. Survey Papers on VRTs and the classification efforts of different VRTs.
ii. Papers introducing new but generally model specific estimators with lower variances.
iii. Papers comparing different VRTs and their combinations in terms of the implementation in different systems and discussing the related results.

A paper by L'Ecuyer [16] consists of an overview of the main techniques in VRT literature and gives some examples regarding those techniques. In general, main techniques existing in the literature can be summarized as follows: Common random numbers, antithetic variates, control variates, importance sampling, indirect estimators, stratification, latin hypercube sampling, conditioning, descriptive sampling, hybrid method, and virtual measures. In addition, some other comprehensive survey papers include Nelson [22] [23], Heidelberger [11], James [12], and Wilson [37].

The first category provides a guide in order to determine the appropriate VRTs and to eliminate the confusion concerning the characteristics of and the relations among VRTs. In this manner, Nelson and Schmeiser [25] present a commendable classification of VRTs. According to their definition, VRTs transform the simulation models into related models that result in more precise estimates of the parameters of interest. These transformations may modify the inputs of a simulation model through distribution replacement or dependence induction as well as may transform the outputs through sample allocation or an equivalent allocation. Also, they may modify the statistics through auxiliary information or equivalent information. These basic transformations later extended in order to make taxonomy of VRTs by Nelson [23] and Nelson and Schmeiser [25]. James [12] approaches this subject in a very different way than Nelson [23], and Nelson and Schmeiser [25] [26] did.

Regarding the second category, Calvin and Nakayama [7] proposed new estimators of some performance measures obtained using regenerative simulations of discrete time markov chain. Moreover, new estimators for the queuing systems are proposed by Law [18].

In the final category, comparisons of different techniques regarding their implementation and efficiency and their combination are taken into account. Nevertheless, these comparisons are performed on very few systems since the techniques are model specific. Therefore, the results obtained via those studies are applicable to related models and the application in other systems may even produce worse results. Also, the comparisons of the single and combined methods are not considered thoroughly in the current literature. The rest will be comprised of a brief summary of these studies.

The first class of the applications and the comparison of different VRTs include the comparison of the single techniques. Glynn and Whitt [10] investigate the asymptotic efficiency of estimators time average queue length L and average waiting time W. They show that an indirect estimator for L using the natural estimator for W and the arrival rate λ is more efficient than a direct estimator for L. This is based on the assumption that the interarrival and waiting time in queue are negatively correlated. Furthermore, they indicate that the indirect or direct estimation is related to the estimation using nonlinear control variables.

Carson and Law [8] focus on the efficient estimation of mean delay in queue, mean wait in system, time average number in queue, time average number in system, and time average amount of work in system for simulated queuing systems of GI/G/s for s=1, 2, and 4. Moreover, Law [18] compares the six efficient estimators for queuing system simulations in terms of efficiency through their variance of the asymptotic distribution of the estimator and variance reduction. He indicated that it is more efficient to estimate the performance means using an estimate of mean delay or waiting time in queue than estimating them directly while the latter is more efficient in single server queuing systems. In addition, based on his empirical studies, using the former estimator is more efficient for GI/G/1 queues whereas the latter seems to perform better in GI/G/2 systems.

Minh [21] proposes the partial conditional expectation technique derived from the conditional expectation technique. He points out that the estimator in this technique is consistent, unbiased and its variance is smaller compared to the crude estimator. Even though the variance reduction achieved by the new technique may not be as much as by conditional expectation but its applicability to some problems to which conditional technique is difficult to apply makes it advantageous.

Cheng [9] compares two different applications of antithetic variates against independent runs for two examples. Then, he finds out that the application of antithetic variates could be very effective if used appropriately. Sullivan et al. [35] investigate the efficiency of the antithetic variate simulation for estimating the expected completion time of the stochastic activity networks. They figure out that antithetic variate could produce the same precision as Monte Carlo simulation but with approximately ¼ computational effort. Avramidis and Wilson [4] propose multiple sample quantile estimators based on antithetic variates and latin hypercube sampling. In addition, the results of the simulation yielding significant reduction in bias and variance are given while the estimation of the quantiles of a stochastic activity networks.

Ahmed et al [1] show that using infinite source and ample server models as control variates for finite source finite server models can be effective in reducing the variance of sensitivity estimates like gradients and Hessians in repairable item systems. Bauer et al. [5] propose a new procedure in order to use control variates in multi-response simulation if the covariance matrix is known. They also present the results of the applications to closed queuing networks and stochastic activity networks.

Wilson and Pritsker [38] give empirical results on the amount of the variance reductions for queuing simulations when the control variates and poststratified sampling are used separately. According to their results, for analytically tractable models of closed and mixed machine repair systems, posstratification produces variance reduction between 10% and 40% and reductions in the half-length between 1% and 20%. In the control variates case, these reductions are between 20% and 90% for variance and 10% and 70% for half-length. Saliby [30] focuses on descriptive sampling that is stated as an improvement over latin hypercube sampling. They gave an example of the application and the comparison of two methods and indicated that descriptive sampling gave better results. Ross [29] indicates how certain VRTs can be efficiently employed during the analysis of the queuing models. He considered three techniques: dynamic stratified sampling, utilization of multiple control variates, and the replacement of random variables by their conditional expectations.

After that, the literature concerning the application of integrated VRTs. Schruben and Mangolin [32] provides the conditions under which the techniques of antithetic variates and common random numbers produce guaranteed efficiency improvements. Kleijnen [14] combines the antithetic variates and common random numbers in order to compare the two alternative systems. He indicated that some implementations of the combined technique could be inferior to antithetic variates or common random numbers individually. Then, he proposes a new combination scheme and experimentally shows the superiority of his scheme to either technique used alone.

Yang and Liou [40] show that the integrated control variates estimated with antithetic variates results in unbiased with smaller variance than the conventional control variate estimator applied without Antithetic Variates. They also prove that the proposed integrated estimator is optimal among the three integrated estimators including the proposed integrated estimator. Kwon and Tew [15] present three methods to combine antithetic variates and control variates which are based on whether the control variates or non-control variates are generated with antithetic variates or not. They indicated that the combined method III, which induces negative correlation among all control and non-control variates, was optimal. Burt and Gaver [6] combine antithetic and control variates and experimentally observed that the combination gave better than the either method applied individually.

Nelson [24] discusses the efficiency of the control variates and antithetic variates in improving the performance of point and interval estimators in the presence of bias due to the determination of the initial bias. Tew and Wilson [36] incorporated control variates into antithetic variates and common random numbers scheme and investigated the conditions under which the combination scheme performed better than antithetic variates and common random numbers, control variates used alone, and direct simulation.

Avramidis and Wilson [2] examined the all pairings of conditional expectation, correlation induction (antithetic variates and latin hypercube sampling) and control variates. Also, they established the sufficient conditions under which that strategy would yield a smaller variance than its constituents would yield individually. They experimented with the stochastic activity networks and indicated that the integrated technique of conditional expectation and latin hypercube sampling performed best.

In this part, some general studies about the VRTs and their integration have been considered. Nevertheless, in the next chapter, we will also mention about the papers specific to each technique.

According to above overview, in general, the literature about the integrated methods is not so intense and as seen most of the studies are devoted to specific models and specific VRTs. Therefore, in this research, we will focus on two other systems (Serial Line Production System and (s,S) Inventory Policy in addition to simple M/M/1 system) and examine the results of the four methods (Antithetic Variates, Latin Hypercube Sampling, Control Variates, and Poststratified Sampling) on these systems. Our study can be classified as in the third category, which includes the researches comparing different VRTs and their combinations in terms of the implementation in different systems and discussing the related results. In this manner, we select those four methods since they were not analyzed thoroughly on different systems.

Our study examines the behaviour of the stand-alone and integrated applications of these methods on one commonly (M/M/1) and two uncommonly used (serial line production system and (s,S) inventory policy) systems. We investigate the reasons behind the success or failure of a technique in a specific system and why the combined techniques perform better than the stand-alone applications. 10 different experiments are performed during the applications of the methods individually or in combination to a system in order to increase the reliability of our conclusions. With these 10 experiments, we construct a confidence interval for the average performances of the techniques.

Chapter 2

Variance Reduction Techniques (VRTs)

The four methods used in this research are Antithetic Variates (AV), Latin Hypercube Sampling (LHS), Control Variates (CV), and Poststratified Sampling (PS). These methods, which are applicable to the simulation of a single system, can be classified into two groups:

1. *Methods that try to induce a correlation among the simulation runs in order to reduce the variance*: AV and LHS. In addition, these methods can be considered as dealing with the input part of the simulation.
2. *Methods that use the auxiliary variables for variance reduction*: CV and PS. Moreover, CV and PS can be considered as dealing with the output part of the simulation.

During the simulation of a Single System, we just try to increase the precision of the estimated performance measure.

2.1. Antithetic Variates (AV)

Antithetic Variates (AV) modifies the input variables in order to reduce the variance. In other words, AV tries to reduce the variance through inducing a negative correlation by using complementary random numbers among the replications. If U_k is a particular random number (uniform between 0 and 1) used for a particular purpose in the first run, we use $1-U_k$ for the same purpose in the second run. That is, a “small” observation in the first run tends to be offset by a “large” observation in the second run.

In this manner, we double the number of runs and average the consecutive odd and even numbered replications to get a smaller variance. As indicated more clearly in the Table 2.1, if 0.153 is used to generate a service time in the first replication, 1-0.153=0.847 is used in the second replication to generate the same service time. By this way, we make the first and the second replications correlated while no dependence occurs with the other pairs.

Rep 1	Rep 2	Rep 3	Rep 4	…	Rep 2n-1	Rep 2n
0.153	0.847	0.697	0.303	…	0.326	0.674
0.631	0.369	0.159	0.841	…	0.455	0.545
0.741	0.259	0.342	0.658	…	0.112	0.888
…	…	…	…	…	…	…
$X_1^{(1)}$	$X_1^{(2)}$	$X_2^{(1)}$	$X_2^{(2)}$	…	$X_n^{(1)}$	$X_n^{(2)}$
X_1		**X_2**		…	**X_n**	

Table 2.1 *Illustration of the Uniform Random Numbers used in AV*

As illustrated better in the Figure 2.1, the desired performance measure is estimated for each pair of runs, which are $(X_j^{(1)}, X_j^{(2)})$.

Negatively Correlated Replications

$$\begin{array}{llllll}
\text{Pair } \mathbf{1}: & X_1^{(1)} & X_1^{(2)} & \Longrightarrow & X_1 & \\
\text{Pair } \mathbf{2}: & X_2^{(1)} & X_2^{(2)} & \Longrightarrow & X_2 & \text{Independent} \\
\dots & \dots & \dots & & \dots & \\
\text{Pair } \mathbf{n}: & X_n^{(1)} & X_n^{(2)} & \Longrightarrow & X_n & \\
 & & & & \overline{X}(n) &
\end{array}$$

Figure 2.1 *Illustration of the Negatively Correlated Replications in AV*

Then, taking the average of the two runs in each pair, we calculate the corresponding average performance measures, X_j's, for each pair:

$$X_j = \frac{X_j^{(1)} + X_j^{(2)}}{2}$$

There exist no dependence between the pairs, hence, replications in different pairs and the averages of the pairs are perfectly independent. Within each pair, complementary random numbers are used for the same purpose by performing synchronization appropriately. More specifically, we use U 's in the $X_j^{(1)}$ for a specific purpose and $1-U$'s for the same purpose in the $X_j^{(2)}$.

After that, $(X_1, X_2,.., X_n)$ can be assumed to be generated from n independent replications and thus the same procedure as in the independent case is followed in order to construct a confidence interval. Assume that the average of X_j 's is $\overline{X}(n)$. Hence, $\overline{X}(n)$ is an unbiased estimator of the desired performance measure:

$$\mu = E(X_j^{(i)}) = E(X_j) = \overline{X}(n)$$

Consequently, $100(1-\alpha)\%$ confidence interval around the mean $E(\overline{X}(n))$ is constructed as follows:

$$\overline{X}(n) \pm t_{n-1,1-\alpha/2} \cdot \sqrt{Var(\overline{X}(n))}$$

Degrees of freedom should be taken as $n-1$ where n is the total number of macro replications while constructing the confidence interval.

The mathematical basis for this method is based on a very simple statistical formula:

$$Var[\overline{X}(n)] = \frac{Var(X_j)}{n} = \frac{Var(X_j^{(1)}) + Var(X_j^{(2)}) + 2Cov(X_j^{(1)}.X_j^{(2)})}{4n} = \frac{Var(X_j^{(1)})}{2n}(1+\rho),$$

where ρ is the correlation between the negatively correlated replications. As it can be seen in the above formula, if the two runs within a pair are made independently, then covariance part will be zero, however, if negative correlation can be induced between the two random variables then covariance part will be negative. Thus the overall variance will be reduced as intended by AV. In another perspective, based on the last part of the above formula, antithetic variate estimator based on n pairs of simulation runs should be more precise than the classical Monte Carlo estimator based on $2n$ independent replications when $\rho < 0$.

Synchronization (i.e., usage of 1-U's for the same purpose U's used) of the input random variables becomes an important issue during the application of AV. Otherwise, the benefit due to the intended negative correlation among the replications may be lost. Law and Kelton [19] mentions some programming tricks about the synchronization of the random variables such as random-number stream dedication, using the inverse-transform method of variate generation wherever possible, judicious wasting of random numbers, pre-generation, and advancing the stream numbers across multiple replications.

Improper synchronization of the input random numbers can cause an increase in the variance of the desired performance measure. Law and Kelton [19] state that unless synchronization is provided appropriately, AV can even backfire; variance of the desired performance measure can even increase. Full synchronization of the all input random variables may not be possible in some large simulation models due to the limited number of seeds available in today's most popular simulation packages. Nevertheless, by determining the most crucial variables in the simulation models and inducing negative correlation among those variables, AV can still be a very useful tool to reduce the variance. A very simple guideline to create negatively correlated replications could be to choose the input variables that affect the desired performance measure significantly. These can be determined through some pilot runs of the simulation model beforehand.

Although AV can be very effective in some cases, its effectiveness is not always guaranteed; we do not know whether AV will reduce the variance and if so, how much the reduction will be. In order to apply AV successfully, the output or response variable is required to be a monotone function of the random variables used to produce the random variates. If the response variable is a monotone function of its arguments, then as shown in Ross [28], the covariance between the complementary replications will be negative. Hence, this negative correlation will help to reduce the variance. In this context, therefore, the Inverse Transform Method is suggested to generate the random variates whenever possible. Using this method, the random variables become a monotone function of the uniform random variates and this is theoretically sufficient for AV to provide a reduction in the variance. For instance, in M/M/1 system since interarrival and service times can be generated using Inverse Transform Method, AV can be effective in reducing the variance of the Time-in-system statistics. However, the magnitude of the reduction still cannot be predicted in finite sample cases in advance.

Example 1. Antithetic Variates (AV)

Consider the simple M/M/1 system with an arrival rate of 9 per minute and service rate of 10 per minute. Suppose that we want to get an estimate of the expected Time-in-system statistics using simulation and we desire the estimator as precise as possible. Using independent sampling with 20 replications, we get the half-length as **0.0547** (by using the estimates of the mean and the standard deviation and applying the common formula for confidence interval with appropriately chosen degrees of freedom), which is not precise as we desire. Hence, we may use a variance reduction technique in order to increase the precision. In this case, 20 replications, of which consecutive odd and even numbered replications are correlated, are taken in order to get 10 independent replications. Negative correlation is induced between both the interarrival and service times. After that we take the average of time-in-system statistics in those correlated replications as indicated in the Table 2.2:

Rep #	ODDS	Time-in-system	Rep #	EVENS	Time-in-system	AVERAGE	Average Time-in-system
1	X_{11}	0.927	2	X_{12}	1.031	X_1	(0.927+1.031)/2= **0.9790**
3	X_{21}	0.907	4	X_{22}	1.048	X_2	(0.907+1.048)/2= **0.9775**
5	X_{31}	0.982	6	X_{32}	1.136	X_3	(0.982+1.136)/2= **1.0590**
7	X_{41}	1.038	8	X_{42}	0.945	X_4	(1.038+0.945)/2= **0.9915**
9	X_{51}	0.899	10	X_{52}	0.907	X_5	(0.899+0.907)/2= **0.9030**
11	X_{61}	1.028	12	X_{62}	0.991	X_6	(1.028+0.991)/2= **1.0095**
13	X_{71}	0.925	14	X_{72}	0.992	X_7	(0.925+0.992)/2= **0.9585**
15	X_{81}	0.979	16	X_{82}	0.941	X_8	(0.979+0.941)/2= **0.9600**
17	X_{91}	1.026	18	X_{92}	0.935	X_9	(1.026+0.935)/2= **0.9805**
19	X_{101}	1.098	20	X_{102}	0.888	X_{10}	(1.098+0.888)/2= **0.9930**

Table 2.2 *An Example for AV*

According to these values, $\hat{V}ar(X_{i1}) = 0.0044$, $\hat{V}ar(X_{i2}) = 0.0056$, and $\hat{C}ov(X_{i1}, X_{i2}) = -0.001834$. Using these data and the knowledge from the statistics,

$$\overline{X} = \frac{1}{10} \times (0.9790 + 0.9775 + ... + 0.9805 + 0.9930) = 0.9812$$

$$\hat{V}ar(\overline{X}) = \frac{\hat{V}ar(X_{i1}) + \hat{V}ar(X_{i2}) + 2 \times \hat{C}ov(X_{i1}, X_{i2})}{4 \times n} \Rightarrow$$

$$\hat{V}ar(\overline{X}) = \frac{0.0044 + 0.0056 + 2 \times (-0.001834)}{4 \times 10} = 0.00016$$

a 95% confidence interval around the mean is ($t_{9,\ 1-0.025}$=2.26):

$$0.9812 \mu\, t_{9,1-0.025} \times \sqrt{0.00016} = 0.9812 \mu\, 0.0286 = [0.9526;1].$$

As a result, reduction in CI compared to independent sampling case is

$$\frac{0.0547 - 0.0286}{0.0547} = 47.71\%$$

2.2. Latin Hypercube Sampling (LHS)

LHS is also based on a correlation induction scheme; it generates negatively correlated micro replications in order to get a single macro replication. This idea is actually similar to AV and LHS can be considered a more general form of AV. AV uses only two micro replications to get one macro replication while LHS can use more than two negatively correlated micro replications. LHS can be considered as a transformation to redefine the inputs of a simulation model by inducing dependence. It works in a very similar way to AV. The average of k output variables from correlated micro replications is used to construct confidence intervals.

Synchronization of the input random numbers is necessary to obtain correlated micro replications.

LHS generates the correlated micro replications that are induced in d dimensions according to the following formula:

$$U_j^{(i)} = \frac{\pi_j(i) - 1 + U_{ij}}{k}$$

for $i = 1,...,k$ and $j = 1,...,d$, where $\pi_1(.),...,\pi_d(.)$ are permutations of the integers $\{1,2,...,k\}$ that are randomly sampled with replacement from the set of $k!$ such permutations. $\pi_j(i)$ denotes the ith element in the j'th sampled permutation and U_{ij} values are uniform random numbers between 0 and 1. Also, U_{ij} is the j'th input random number used in the i'th replication. As seen in the above formula and the table below, LHS wastes many random numbers as compared to AV and thus it requires much more time and effort. In general, the input random numbers in LHS can be represented as in Table 2.3.

Entity Number	**Micro Replication 1**	**Micro Replication 2**	**Micro Replication 3**	**…**	**Micro Replication k**
1	$\frac{(\pi_1(1)-1+U_{11})}{k}$	$\frac{(\pi_1(2)-1+U_{21})}{k}$	$\frac{(\pi_1(3)-1+U_{31})}{k}$	…	$\frac{(\pi_1(k)-1+U_{k1})}{k}$
…	…	…	…	…	…
j	$\frac{(\pi_j(1)-1+U_{1j})}{k}$	$\frac{(\pi_j(2)-1+U_{2j})}{k}$	$\frac{(\pi_j(3)-1+U_{3j})}{k}$	…	$\frac{(\pi_j(k)-1+U_{kj})}{k}$
…	…	…	…	…	…
d	$\frac{(\pi_d(1)-1+U_{1d})}{k}$	$\frac{(\pi_d(2)-1+U_{2d})}{k}$	$\frac{(\pi_d(3)-1+U_{3d})}{k}$	…	$\frac{(\pi_d(k)-1+U_{kd})}{k}$

Table 2.3 *Illustration of the Uniform Random Numbers used in LHS*

The essence of LHS is the independent generation of random permutations and uniform random numbers. In this manner, for each random number, a stratified sample of size k is taken from a uniform distribution on [0,1] so that observations within the sample and observations in each stratum are negatively correlated. LHS generates the random variates using those negatively correlated random numbers and the Inverse Transform Method.

In LHS, the region between 0 and 1 is divided into k non-overlapping intervals of equal length for each random variable. Hence, k different values in the k non-overlapping intervals are selected randomly for each random variable so that one value from each interval is generated. In this way, we ensure that the input random variable X has all portions of its distribution. Hence, if an input random variable in one replication uses a uniform number from a region for a specific purpose, then in other negatively correlated replications, same input random variable is forced to use another uniform random number from the other regions. For $i = 1,...,k$, i'th region can be considered as

$$R_i = \left(\frac{i-1}{k}, \frac{i}{k} \right]$$

which simply follows from the method of generating uniform random numbers in LHS.

In this method, we take $n.k$ replications in total as k micro replications are taken for one macro replication. Then, we calculate average of each macro replication using the averages of the micro replications. As illustrated in the Figure 2.2, the desired performance measure is estimated for each pair of runs, which are $(X_j^{(1)}, X_j^{(2)},..,X_j^{(k)})$.

$$\begin{array}{l} \\ \text{Macro Replication } \mathbf{1}\text{:} \\ \text{Macro Replication } \mathbf{2}\text{:} \\ \cdots \\ \text{Macro Replication } \mathbf{n}\text{:} \end{array}
\begin{array}{c} \overbrace{\qquad\qquad\qquad\qquad\qquad\qquad}^{\textbf{Micro Replications (Negatively Correlated)}} \\ \begin{array}{cccc} X_1^{(1)} & X_1^{(2)} & \cdots & X_1^{(k)} \\ X_2^{(1)} & X_2^{(2)} & \cdots & X_2^{(k)} \\ \cdots & \cdots & \cdots & \cdots \\ X_n^{(1)} & X_n^{(2)} & \cdots & X_n^{(k)} \end{array} \end{array}
\begin{array}{c} \\ \Longrightarrow \\ \Longrightarrow \\ \\ \Longrightarrow \end{array}
\begin{array}{c} \\ \left.\begin{array}{c} X_1 \\ X_2 \\ \cdots \\ X_n \end{array}\right\} \textbf{Independent} \\ \hline \bar{X}(n) \end{array}$$

Figure 2.2 *Illustration of the Negatively Correlated Replications in LHS*

Then, taking the average of the each micro replication's average corresponding to each macro replication, we calculate the corresponding average performance measures, X_j's, for each macro replication:

$$X_j = \frac{X_j^{(1)} + X_j^{(2)} + ... + X_j^{(k)}}{k}$$

Since the selection of the permutations and the uniform random numbers while determining the uniform random number to be used in micro replications are performed randomly, there does not exist any dependency among the micro replications corresponding to different macro replications, hence, micro replications belonging to separate macro replications and thus their averages are perfectly independent.

After that, $(X_1, X_2,..,X_n)$ can be assumed to be generated from n independent replications since they are perfectly independent and thus the same procedure as in the independent case is followed in order to construct a confidence interval. Assume that the average of X_j 's is $\overline{X}(n)$. Hence, $\overline{X}(n)$ is an unbiased estimator of the desired performance measure, $\mu = E(\overline{X}(n))$ and $\mu = E(X_j)$. Consequently, $100(1-\alpha)\%$ confidence interval around the mean $E(\overline{X}(n))$ is constructed as follows:

$$\overline{X}(n) \pm t_{n-1,1-\alpha/2} \cdot \sqrt{Var(\overline{X}(n))}$$

While constructing the confidence interval, degrees of freedom should be taken as $n-1$ where n is the total number of macro replications.

In fact, LHS is proposed to be more effective than direct simulation provided that the output variable is a monotone function of the input random variables. One advantage of LHS occurs when the output random variable is dominated by a few of input random variables. In this case, applying LHS on all input dimensions assures us that all input variables are represented in a fully stratified manner.

McKay, Beckman, and Conover [20], the originators of LHS, proved that the LHS estimators are unbiased in their original papers. Furthermore, Stein [34] proved that the variance of the estimator is lower compared to simple random sampling. When LHS was firstly proposed by McKay, Beckman, and Conover [20], they assumed that the input random variates were independent and each of these variates was generated using the Inverse Transform method. On the other hand, Avramidis and Wilson [4] generalized LHS so that none of these assumptions were required anymore.

Example 2. Latin Hypercube Sampling (LHS)

Consider the simple M/M/1 system with an arrival rate of 9 per minute and service rate of 10 per minute. Using 30 independent replications, we construct a confidence interval with a half-length of **0.0494**. Since *d* is the number of entities in each simulation run, it is equal to 10,000 in this case. In this example, we will use Latin Hypercube Sampling in order to reduce the variation around the mean. In order to induce negative correlation around the mean, we will produce service times using negatively correlated uniform random numbers and inverse transformation method. This can be summarized as follows:

- Determine the number of correlated replications or level of stratification (*k*), say *k=3*.
- Generate a random permutation of the numbers: 1, 2, 3 (*k*), say 3-1-2.
- Generate three uniform random numbers, say $\mathbf{u_1}$=0.5470, $\mathbf{u_2}$=0.3450, and $\mathbf{u_3}$=0.9365, respectively.

- For micro replication 1, use $\mathbf{U_1}=(i\text{-}1+\mathbf{u_1})/k=(3\text{-}1+0.5470)/3=$ 0.849 to generate an observation. Similarly, use $\mathbf{U_2}=(i\text{-}1+\mathbf{u_2})/k=(1\text{-}1+0.3450)/3=$ 0.1150 for micro replication 2 and $\mathbf{U_3}=(i\text{-}1+\mathbf{u_3})/k=(2\text{-}1+0.9365)/3=$ 0.6455 for micro replication 3.
- Using the above procedure, generate the observations in the simulation run length and collect the required statistics, which is time-in-system in our case.
- Calculate the average of the averages of those three correlated micro replications to calculate the corresponding time-in-system value for each macro replication.
- After that follow the same procedure in the independent case to calculate the confidence interval and take the degrees of freedom as $n\text{-}1$=10-1=9.

Basic idea of using the above formula to generate input uniform random numbers for micro replications is to choose observations so that while one of the observations is in the low level, one of the other will be in the middle level and the other will be in the high level. That is, if a small service time is generated in the micro replication 1, a large service time is generated in one of the other two replications and a nearly average service time is generated in the last. As illustrated for our case in the Figure 2.3, [0, 1] length is divided to three equal parts and the observations or random variables in the micro replications are forced to use only one of these parts while generating an observation. In this figure, we show that if the permutation turns out to be 1, the input uniform random number is enforced to be in the first interval. The same routine is followed for other permutation values.

$$i=1 \Rightarrow U_1 = \frac{1-1+U}{3} = \frac{U}{3} \Rightarrow 0 \le U_1 \le \frac{1}{3} \Rightarrow U_1 \in R_1$$
$$i=2 \Rightarrow U_1 = \frac{2-1+U}{3} = \frac{1}{3}+\frac{U}{3} \Rightarrow \frac{1}{3} < U_1 \le \frac{2}{3} \Rightarrow U_1 \in R_2$$
$$i=3 \Rightarrow U_1 = \frac{3-1+U}{3} = \frac{2}{3}+\frac{U}{3} \Rightarrow \frac{2}{2} < U_1 \le 1 \Rightarrow U_1 \in R_3$$

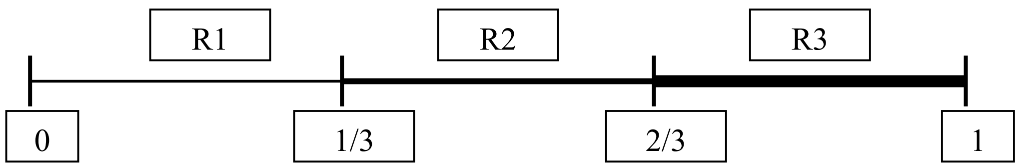

Figure 2.3 *Illustration of the Stratification Idea in LHS*

In the Table 2.4, basic mechanism of this procedure is illustrated better for our case. We generate 40,000 uniform numbers and use the given formula to produce the random numbers that will be used to generate the service times in each of three micro replications.

	Uniform R.V.(**U**)	Random Permutation	Micro Replication 1	Micro Replication 2	Micro Replication 3
Entity No 1	**U_1**=0.5470 **u_2**=0.3450 **u_3**=0.9365	3-1-2	(**3**-1+**0.5470**)/3 = 0.8490	(**1**-1+**0.3450**)/3 = 0.1150	(**2**-1+**0.9365**)/3 = 0.6455
Entity No 2	**u_1**=0.1180 **u_2**=0.6424 **u_3**=0.4382	1-3-2	(**1**-1+**0.1180**)/3 = 0.0393	(**3**-1+**0.6424**)/3 = 0.8808	(**2**-1+**0.4382**)/3 = 0.4794
...	...	...	...	...	...
Entity No 40000	**u_1**=0.8760 **u_2**=0.2791 **u_3**=0.7542	2-3-1	(**2**-1+**0.8760**)/3 = 0.6253	(**3**-1+**0.2791**)/3 = 0.7597	(**1**-1+**0.7542**)/3 = 0.2514

Table 2.4 *Illustration of the Implementation of LHS*

After running the model, we collect the time-in-system statistics for each micro replication as shown in the Table 2.5. Since we generate $k=3$ correlated micro replications in order to get one macro replication, 10x3=30 micro replications have been taken for 10 macro replications:

i ***j**=1,2,3*	Macro Replication 1	Macro Replication 2	Macro Replication 3	Macro Replication 4	Macro Replication 5	Macro Replication 6	Macro Replication 7	Macro Replication 8	Macro Replication 9	Macro Replication 10
Micro ***i*** 1	1.07590	0.94422	0.98221	0.93811	0.95893	0.91587	1.03530	0.97252	0.90675	1.03900
Micro ***i*** 2	1.00570	1.07700	1.07030	1.03520	0.84605	1.14150	0.95079	1.08260	0.87119	1.02330
Micro ***i*** 3	0.98294	0.93460	0.84025	1.05600	0.93333	0.99405	1.13620	0.86528	0.89788	0.95832
Average	**X_1=1.022**	**X_2=0.985**	**X_3=0.964**	**X_4=1.010**	**X_5=0.913**	**X_6=1.017**	**X_7=1.041**	**X_8=0.973**	**X_9=0.892**	**X_{10}=1.007**

Table 2.5 *An Example for LHS*

After calculating the average time-in-system statistics for each macro replication, we use the same procedure with the independent sampling case and use degrees of freedom as 10-1=9.

$$\overline{X} = \frac{1}{10} \times (1.022 + 0.985 + ... + 1.007) = 0.9824$$

$$\hat{V}ar(X) = \frac{1}{9} \times \sum (X_i - 0.9824)^2 = (0.0482)^2$$

a 95% confidence interval around the mean is

$$0.9824 \mu\, t_{9,1-0.025} \times \sqrt{\frac{(0.0482)^2}{10}} = 0.9824 \mu\, 0.0345 = [0.9479, 1.0168]$$

As a result, reduction in CI compared to independent sampling case is

$$\frac{0.0494 - 0.0345}{0.0494} = 30.16\%$$

2.3. Control Variates (CV)

CV method incorporates prior knowledge by the usage of a secondary variable, hence takes advantage of correlation between random variables in order to reduce the variance. It does not change the simulation run in any way and in fact, any simulation includes potential control variates. This makes the usage of CV feasible for any stochastic simulation experiment. In fact, CV assumes a linear correlation between the output random variable and the control variate and thus this method is proposed for systems with a strong linear association between the output random variable and the control variate. Unlike the most other variance reduction techniques, this correlation can be either negative or positive. One of the fundamental requirements of the CV is to choose a control variate, which has a known mean, μ_Y, before the simulation.

Assume that we try to estimate the average time-in-system in a simple M/M/1 system. In this system, we have two system parameters: interarrival time and service time. Prior to the simulation, we know the theoretical means of these parameters. Time-in-system has two components that are waiting time in queue and service time. Hence, we may use the service time as a control variate while estimating the average time-in-system since we know the theoretical mean of service time beforehand. Intuitively, existence of a positive correlation between the service time and the average time-in-system can be realized easily since the total time spent in the system is the sum of waiting time in queue and the service time. If a large (small) service time occurs for one customer in the M/M/1 system, then we expect a large (small) time-in-system for that customer. Accordingly, we adjust the value of time-in-system upward or downward depending on whether we observed a large or small service time, respectively. As a result, we bring the observed time-in-system values towards its mean and thus make its variance smaller. In fact this is the basic idea behind the CV method.

Consider now a serial production line and assume that we want to estimate the throughput of the system. Like the previous case, we may choose the processing time of a machine or station, which has a known mean value prior to the simulation, as a control variate. Contrary to the previous case of M/M/1 system, there exists a negative correlation between the throughput of the production line and the processing time of the machine. If a large processing time is observed in a replication, then we expect a small throughput value for the system and vice versa. According to the observed values of machine processing times, we adjust the observed value of throughput for each replication and make them closer to the actual mean of the throughput. That is, when we obtain a large (small) processing time, we expect a small (large) throughput from the simulation and adjust this observed throughput value downward (upward). As illustrated with two examples, the sign of the correlation is not important in CV method.

An obvious problem appears with the CV method: What will be the amount of the adjustments? Firstly, let us denote the output random variable to be estimated with X and the control variate having a known mean value with Y. Then, we express the deviation of the control variate from its known mean value as $Y-\mu$ and multiply this value with a constant coefficient a in order to scale the deviation $Y-\mu$ to make an adjustment to X. Hence, we state the "controlled" or "corrected" estimator of X as follows:

$$X_c = X - a(Y - \mu_y)$$

This expression reveals that positive correlation between X and Y makes the value of a positive. In that case, observed value of $Y > \mu_y$ pulls the corresponding X value downward and vice versa, making it closer to $\mu = E(X)$. However, since a negative correlation between X and Y causes a to be negative, observed value of $Y > \mu_y$ pulls the corresponding X value upward.

Since $E[Y] = \mu_Y$, X_c is an unbiased estimator of $\mu = E(X)$, which is, in fact, expected to have a lower variance than X. Using the basic statistics knowledge, $Var(X_c)$ can be written as follows:

$$Var(X_c) = Var(X) + a^2 Var(Y) - 2aCov(X,Y)$$

Of course, whether the variance of corrected estimator will be smaller than the original estimator totally depends on the choice of Y and the value of a. However, the optimal value of a for a chosen Y can be calculated by taking the derivative of the above formula over a.

$$\frac{dVar(X_c)}{da} = 2aVar(Y) - 2Cov(X,Y) = 0 \Rightarrow a^* = \frac{Cov(X,Y)}{Var(Y)}$$

Furthermore, second derivative of $Var(X_c)$ over a is $2Var(Y) > 0$, hence, a^* minimizes the variance of the corrected estimator, $Var(X_c)$.

According to the optimal value of a^*, if there exists a strong correlation between the variables X and Y, then the covariance part will be large and thus a large a^* value will be used in calculating the corrected variable. This means that the deviations of Y from its known mean will tell us more about the deviation of X from its true mean, hence, we will make adjustments more confidently on X. In other words, for a selected control variate Y, the larger the value of a^*, the more the variance reduction there will be. Nevertheless, the values of a^* belonging to different control variates should not be compared with each other since this value depends also on the scale differences between X and the chosen control variate. Only the correlation between X and Y's should be used while predicting and comparing the effectiveness of different control variates. In addition, as we make the denominator of $a*$, $Var(Y)$, smaller, then a large $a*$ is obtained, which increases the variance reduction.

Substituting the a value with its optimal $a*$ in the $Var(X_c)$ formula, the following is obtained:

$$Var(X_c^*) = Var(X) - \frac{Cov^2(X,Y)}{Var(Y)} = (1-\rho_{XY}^2)Var(X),$$

where ρ_{XY}^2 is the correlation coefficient between X and Y. Actually, this formulation indicates the mathematical basis for the CV method. Using the optimal value $a*$, the variance of the controlled or corrected estimator can never be larger than the variance of the original estimator, X. Besides, as the correlation between X and Y increases, $Var(X_c)$ goes to zero, which says that as $\rho_{XY}^2 \to 0$, almost all the time we adjust the X value to its true mean and thus getting zero variance.

After that, $(X_c^{(1)}, X_c^{(2)}, .., X_c^{(n)})$ can be assumed to be generated from n independent replications and thus the same procedure as in the independent case is followed in order to construct a confidence interval. Consequently, $100(1-\alpha)\%$ confidence interval around the mean $E(\overline{X}_c(n))$ is constructed as follows:

$$\overline{X}_c(n) \pm t_{n-1,1-\alpha/2} \cdot \sqrt{Var(\overline{X}_c(n))}$$

While constructing the confidence interval, degrees of freedom should be taken as $n-1$ where n is the total number of macro replications if $a*$ is known previously.

The previous discussion is based on the assumption that $Cov(X,Y)$ and $Var(Y)$ are known prior to the simulation. Actually, this is not a very realistic assumption in most practical conditions. Even though we may know the value of $Var(Y)$ in some cases, $Cov(X,Y)$ is generally not known beforehand. Lavenberg, Moeller, and Welch [17] proposed to use the estimates of $Cov(X,Y)$ and $Var(Y)$ in order to estimate $a*$.

Suppose that we have obtained the observations $X_1, X_2, ..., X_n$ and $Y_1, Y_2, ..., Y_n$ from the *n* replications of the simulation model. Assume that $\overline{X}(n)$ and $\overline{Y}(n)$ are the sample means of those observations and $\hat{S}_Y^2(n)$ is the sample variance of the Y_j's. Then, the correlation between X and Y and the corresponding value of $a*$ are estimated as follows:

$$\hat{C}_{XY}(n) = \frac{\sum_{j=1}^{n}\left[X_j - \overline{X}(n)\right]\left[Y_j - \overline{Y}(n)\right]}{n-1} \Rightarrow \hat{a}^*(n) = \frac{\hat{C}_{XY}(n)}{S_Y^2(n)}$$

As a result, the following expression is used to estimate the mean of controlled estimator:

$$\overline{X}_c^*(n) = \overline{X}(n) - \hat{a}^*(n)\left[\overline{Y}(n) - \mu_Y\right]$$

However, since the estimate, $\hat{a}^*(n)$, is used instead of a^*, degrees of freedom should be taken as $n-2$ while constructing the confidence interval. Obviously, $\overline{X}_c^*(n)$ is not an unbiased estimator of $\mu = E(X)$ unlike X_c and X_c^* since in this case, a dependency occurs between $\hat{a}^*(n)$ and $\overline{Y}(n)$. Therefore, we cannot ensure that the corrected or controlled estimator is always unbiased and the estimator has a smaller variance than $\overline{X}(n)$. Lavenberg, Moeller, and Welch [17] discuss the severity of this bias.

In the literature, many studies focusing on the estimation of a^* and the alternative estimators to a^* have been made. Kleijnen [13] used jackknifing to reduce the bias in $\overline{X}_c^*(n)$ as an alternative estimator to a^* and Avramidis and Wilson [3] splits the output data to estimate a^*. Furthermore, Nelson [24] established the following central limit theorem for CV, which states that $\overline{X}_c^*(n)$ asymptotically dominates $\overline{X}(n)$:

$$\frac{(\overline{X}_c^*(n) - \mu_X)}{\sqrt{n}} \to N\left[0, \sigma_X^2(1 - R_{XY}^2)\right] \text{ as } n \to \infty \text{, where}$$

$$R_{XY}^2 = \frac{Cov^2(X,Y)}{Var(X)Var(Y)} = \rho_{XY}^2$$

In its more general form, we define the controlled estimator

$$X_c = X - a(Y - \mu_Y) = X - \sum_{k=1}^{q} a_k (Y^{(k)} - \mu_Y^k),$$

where $a = (a_1, ..., a_q)'$ is a vector of constants. Due to possibility of the correlation between the control variates,

$$Var(X_c) = Var(X) + \sum_{i=1}^{q} a_i^2 Var(Y_i) - 2\sum_{i=1}^{q} a_i Cov(X, Y_i) + 2\sum_{i_1=2}^{q}\sum_{i_2=1}^{i_1-1} a_{i_1} a_{i_2} Cov(Y_{i_1}, Y_{i_2})$$

In this case, we have to solve q equations to calculate the optimal values of a_i's after taking derivatives. Again, problems regarding the bias of the estimator due to the estimated values of a_i's occur in the q control variate case. On the other hand, assuming the control variates as independent, we can calculate the a_i's easily. Let $\sum_Y = Cov[Y]$ be a matrix whose element (i, j) is the value of $Cov[Y^{(i)}, Y^{(j)}]$, and

$$\sigma_{XY} = (Cov(X, Y^{(1)}),..., Cov(X, Y^{(q)}))'.$$

Then, $E[X_c] = E[X] = \mu$ and

$$Var[X_c] = Var[X] + a'\sum_Y a - 2a'\sigma_{XY} .$$

This variance is minimized with $a = a^* = \sum_Y^{-1} .\sigma_{XY}$ in which case

$$Var[X_c] = (1 - R_{XY}^2)Var[X],$$

where

$$R_{XY}^2 = \frac{\sigma'_{XY} \sum_Y^{-1} \sigma_{XY}}{Var[X]}$$

is the coefficient of determination, the square of the multiple correlation coefficient, between X and Y. Nevertheless, the above formulas assume again that a^* is known and this may not always be a valid assumption. If (X, Y) is assumed to be multinormal and a^* is estimated using the sample covariance and variance values, then

$$Var[X_c] = \frac{n-2}{n-q-2}(1 - R_{XY}^2)Var[X]$$

Actually, the term $(n-2)/(n-q-2)$ is called the "loss factor". However, asymptotically, X_c always has a smaller MSE than X and the there is no loss due to the estimation of the a^*. In its general form, if q control variates with the estimated values $\hat{a}^*(n)$ are used, degrees of freedom should be taken as $n - q - 1$.

In addition to unbiased CV estimators, Schmeiser, Taaffe and Wang [31] consider the Biased Control Variates (BCVs) in order to increase the efficiency of stochastic simulation experiments. BCVs uses an approximation for the mean of the control variate so the resulting control variate estimator is biased. This estimator minimized the more general Mean Squared Error (MSE) that is the sum of the estimator variance plus the bias squared.

As stated before, as the correlation between X and Y increases, then we get a smaller variance around the $\mu = E(X)$. Therefore, the choice of control variate carries a vital importance in the application of CV. Law and Kelton [19] classifies the possible control variate sources into three: Internal, External, and Using Multiple Estimators.

Internal control variates refer to the input random variables or their simple functions such as averages. In general, their expectations are known before the simulation and the expected correlation between any of the input random variables and output random variable can be explored using some pilot runs of the simulation or a simple analysis of their contribution to the output random variable. Actually, choosing an input random variable does not add anything to the cost of simulation run since they must be determined in advance. *External* control variates refer to the output random variates, whose means are not known previously but estimated using a second simulation run. This additional run could be the original model itself as well as its simplified version. Hence, selection of an external control variate increases the cost of the simulation.

Sometimes, we may have more than one unbiased estimator $X^{(1)}, X^{(2)},..., X^{(k)}$ for μ and thus we may be *Using Multiple Estimators* to get a variance reduction. If $a_1,..., a_k$ are any real numbers whose sum equals 1, then

$$X_c = \sum_{i=1}^{k} a_i X^{(i)}$$

is also an unbiased estimator of μ. Since $a_1 = 1 - \sum_{i=1}^{k} a_i$, X_c can be expresses as

$$X_c = \left(1 - \sum_{i=1}^{k} a_i\right) X^{(1)} + \sum_{i=2}^{k} a_i X^{(i)} = X^{(1)} - \sum_{i=2}^{k} a_i (X^{(1)} - X^{(i)})$$

Hence, $Y_i = X^{(1)} - X^{(i)}$ for $i = 2,3,...,k$ are $k-1$ control variates for $X^{(1)}$.

Implementation of CV in the simulation of a single system is illustrated better in the following example, which considers CV in its simplest form and chooses only one control variate.

Example 3. Control Variates (CV)

Consider the simple M/M/1 system with an arrival rate of 9 per minute and service rate of 10 per minute again, where we constructed the half-length as **0.0717** using 10 independent replications. Since this method uses the correlation between certain random variables to obtain a variance reduction, firstly, we have to choose a control variate which has a correlation with the desired random variable. In this case, service time has been chosen as the control variate since in the beginning of the simulation we know the theoretical mean of service time and it has a correlation with the time-in-system. This method can be summarized as follows:

- Choose a control variate, **Y**, which has a correlation with the parameter that we want to estimate, **X**, , i.e. service in our case in order to estimate average time-in-system.
- Calculate the control coefficient **a*** as **Cov(X, Y)/Var(Y)**.
- Calculate the controlled response as $\mathbf{X_c=X- a^*.(Y-\mu_y)}$ for each replication.

- Assume that the controlled response is the output of independent replications and follow the same procedure as usual.

In the Table 2.6, both the time-in-system and the waiting time in queue for our case are given:

	Run1	Run2	Run3	Run4	Run5	Run6	Run7	Run8	Run9	Run10
ServiceTime, **Y**	0.100	0.099	0.101	0.100	0.099	0.101	0.099	0.100	0.100	0.101
TimeInSys, **X**	0.990	0.935	1.130	1.169	0.924	1.077	0.833	0.996	1.008	1.047

Table 2.6 *Replication Averages for Control Variate and Response Variable in CV*

According to these values, we estimate the required statistics as, $\overline{X} = 1.0109$, $\hat{V}ar(X) = 0.0101$, $\hat{V}ar(Y) = 0.0000004$, $\hat{C}ov(X,Y) = 0.000055$

$$a^* = \frac{\hat{C}ov(X,Y)}{\hat{V}ar(Y)} = \frac{0.000055}{0.0000004} = 137.5$$

Using this value of control coefficient, we calculate the corrected values of time-in-system values as in Table 2.7:

Corrected Time-in-system Value		Service Time		Time-in-system	
$\mathbf{X_{c1}}$	0.990-137.5*(0.10031-0.1)= **0.948**	Y_1	0.100	X_1	0.990
$\mathbf{X_{c2}}$	0.935-137.5*(0.09900-0.1)= **1.035**	Y_2	0.099	X_2	0.935
$\mathbf{X_{c3}}$	1.130-137.5*(0.10110-0.1)= **0.979**	Y_3	0.101	X_3	1.130
$\mathbf{X_{c4}}$	1.169-137.5*(0.10030-0.1)= **1.128**	Y_4	0.100	X_4	1.169
$\mathbf{X_{c5}}$	0.924-137.5*(0.09938-0.1)= **1.009**	Y_5	0.099	X_5	0.924
$\mathbf{X_{c6}}$	1.077-137.5*(0.10083-0.1)= **0.962**	Y_6	0.101	X_6	1.077
$\mathbf{X_{c7}}$	0.833-137.5*(0.09922-0.1)= **0.940**	Y_7	0.099	X_7	0.833
$\mathbf{X_{c8}}$	0.996-137.5*(0.09983-0.1)= **1.020**	Y_8	0.100	X_8	0.996
$\mathbf{X_{c9}}$	1.008-137.5*(0.09984-0.1)= **1.030**	Y_9	0.100	X_9	1.008
$\mathbf{X_{c10}}$	1.047-137.5*(0.10052-0.1)= **0.976**	Y_{10}	0.101	X_{10}	1.047

Table 2.7 *An Example for CV*

Using the corrected time-in-system values, we calculate the half-length and construct the confidence interval as usual:

$$\overline{X}_c = \frac{1}{10} \times (0.948 + 1.035 + ... + 0.976) = 1.0027$$

$$\hat{V}ar(X) = \frac{1}{9} \times \sum (X_i - 1.0027)^2 = 0.0030827$$

a 95% confidence interval around the mean is ($t_{8,\ 1\text{-}0.025}$=2.306):

$$1.0027 \mu\, t_{8,1-0.025} \times \sqrt{\frac{0.0030827}{10}} = 1.0027 \mu\, 0.0405 = [0.9622;1.0432]$$

As a result, reduction in CI compared to independent sampling case is

$$\frac{0.0717 - 0.0405}{0.0717} = 43.52\%$$

2.4. Poststratified Sampling (PS)

As opposed to LHS, which is based on redefining the input random numbers of a simulation, PS bases the efficiency improvements on auxiliary information extracted from internal system variables just like the Control Variates (CV) method. Wilson and Pritsker [38] firstly proposed this method in variance reduction. As stated in their paper, this method is a result of an investigation of a technique, which

- does not alter the time path generated by a simulation but instead uses the concomitant variables obtained through the application of the simulation in order to reduce the variance, and
- is robust and asymptotically stable in order to use this auxiliary information.

Moreover, they propose this method for systems with a complex nonlinear relation between the response and the concomitant variables. In PS, a poststratified estimator for the expected value μ_Y of the response variable Y requires the determination of an effective standardized stratification variate C, which should be asymptotically standard normal random variable. When the number of strata, L, and the points of stratification are specified, the following stratification scheme is constructed before the simulation experiments:

$$\{\zeta_0 = -\infty, \zeta_1, \zeta_2, \zeta_3, \dots, \zeta_{L=1}, \zeta_L = \infty\}$$

At the end of each replication, the observed value of C is normalized and is used to classify the response Y into appropriate stratum. There is no attempt to force the observations into prespecified strata by manipulating the random number input. In the h'th stratum, the following are defined:

$$\begin{aligned} \pi_h &= P(\zeta_{h-1} < C \le \zeta_h) \\ \mu_{Yh} &= E\left[Y \mid \zeta_{h-1} < C \le \zeta_h\right] \\ \sigma_{Yh}^2 &= E\left[(Y - \mu_{Yh})^2 \mid \zeta_{h-1} < C \le \zeta_h\right] \end{aligned}$$

At the end of the n replications, the number of desired output variables N_h falling in the h'th stratum is a binomial random variable with parameters n and π_h provided that the number of observations in each strata are nonzero, that is, $N_h > 0$ for $1 \le h \le L$. Denoting Y_{hj}, $1 \le j \le N_h$, the response variable falling in the h'th stratum; the poststratified estimator of μ_Y is

$$\overline{Y}_{ps} = \sum_{h=1}^{L} \pi_h \left(\frac{1}{N_h} \sum_{j=1}^{N_h} Y_{hj} \right)$$

The variance of $\overline{Y}_{ps}$ is estimated to order n^{-2} by

$$Var(\overline{Y}_{ps}) = \sum_{h=1}^{L} \frac{1}{N_h} \pi_h^2 S_{yh}^2 ,$$

where S_{Yh}^2 is the sample variance of the observations in the h'th stratum. Stratification of the output variable according to the value of the stratification variate produces a smaller variance in each stratum. Actually, PS combines those smaller variance values in order to reduce the variance.

Suppose that the number of observations in each stratum is positive and let $\overline{Y}_h$ and S_{Yh}^2 are the sample mean and the variance within stratum h, $1 \le h \le L$. In PS, the response or output random variable is assumed to be normally distributed within each stratum h, $1 \le h \le L$; Y_{hj} is $N(\mu_{Y_h}, \sigma_{Yh}^2)$ (Wilson and Pritsker [38]). Then, conditioning on the total number of observations $N = n$, the sample statistics $\left\{\overline{Y}_h, S_{Yh}^2 : 1 \le h \le L\right\}$ are mutually independent and $\overline{Y}_{ps}$ is normally distributed with

$$E\left[\overline{Y}_{ps} \mid N = n\right] = \mu_Y \text{ and } Var\left[\overline{Y}_{ps} \mid N = n\right] = \sum_{h=1}^{L} \frac{1}{N_h} \pi_h^2 \sigma_{yh}^2 .$$

Consequently,

$$\hat{V}ar\left[\overline{Y}_{ps} \mid N = n\right] = \sum_{h=1}^{L} \frac{1}{N_h} \pi_h^2 S_{yh}^2$$

is the conditional variance estimator. In order to construct the confidence interval μ_Y, the effective degrees of freedom should be determined since we find the mean and variance conditioning on the number of observations falling in each stratum. Effective degrees of freedom is calculated by the following formula (Wilson and Pritsker [38]):

$$\upsilon_e = \frac{\left(\sum_{k=1}^{L} \frac{1}{N_h} \pi_h^2 S_{Yh}^2 \right)^2}{\sum_{k=1}^{L} \frac{\pi_h^4 S_{Yh}^4}{N_h^2 (N_h + 1)}} - 2$$

As a result, the following confidence interval has the asymptotic conditional coverage probability $1 - \alpha$ for μ_Y:

$$\left[\overline{Y}_{ps} - t_{1-\alpha/2}(\nu_e \text{ d.o.f.}).\sqrt{\hat{V}ar(\overline{Y}_{ps} \mid N = n)}\, ; \overline{Y}_{ps} + t_{1-\alpha/2}(\nu_e \text{ d.o.f.}).\sqrt{\hat{V}ar(\overline{Y}_{ps} \mid N = n)} \right]$$

The proofs of the above formulas are given in Wilson and Pritsker [38].

The magnitude of the variance reduction that will be obtained using PS depends on number of strata L and the set of standard normal stratification points:

$$\{\zeta_0 = -\infty, \zeta_1, \zeta_2, \zeta_3, \ldots, \zeta_{L=1}, \zeta_L = \infty\}$$

Wilson and Pritsker [38] propose the usage of $2 \leq L \leq 6$ strata taking the relation between L and the variance of the stratified sample mean. Furthermore, assuming the linear regression relation $Y = \beta_0 + \beta_1 C + e$ holds with $E[e] = 0$ and $Var(e) = \sigma^2$, they calculated the points of stratification corresponding to each value of $2 \leq L \leq 6$. Table 2.8 illustrates the stratification scheme by Sethi [33] for a standard normal variate under PS:

Number of Strata	Upper Limit of Stratum h, Weight of Stratum h,					
	Stratum, h					
L	1	2	3	4	5	6
2	0 0.5	Infinity 0.5				
3	-0.612 0.2705	0.612 0.459	Infinity 0.2705			
4	-0.982 0.163	0 0.337	0.982 0.337	Infinity 0.163		
5	-1.244 0.107	-0.382 0.244	0.382 0.298	1.244 0.244	Infinity 0.107	
6	-1.447 0.074	-0.659 0.181	0 0.245	0.659 0.245	1.447 0.181	Infinity 0.074

Table 2.8 *Sethi's Optimal Stratification Scheme for a Standard Normal Variate*

Example 4. Post-Stratification (PS)

Consider the M/M/1 case in CV, where we obtained the half-length as **0.0717** using 10 independent runs. Now, we consider the Post-stratified sampling to decrease the variation around the mean of time-in-system statistics. This method can be summarized as follows:

- Determine the standardized stratification variate, which is service time in our case.
- Determine the number of replications and the level of stratification, which are n=10 and L=3 respectively in our case. Take n replications.
- Construct each stratum according to the optimal stratification scheme proposed by Sethi [33]. During this construction procedure, use the theoretical mean and the standard deviation of the stratification variate.
- Place the corresponding time-in-system (response variable) values into these strata according to corresponding service time value.
- Calculate the mean and the variance of time-in-system in each stratum using the related formulas.
- Calculate the effective degrees of freedom using the given formula.
- Construct the confidence interval as usual.

In this case, we take 10 replications in the independent case but this time hold the value of ‘service time’ in each replication, which we choose as the standardized stratification variate. Moreover, we specify the number of strata as *3.* Table 2.9 indicates the values of time-in-system and service time for each replication:

Service Time, **X**	0.099	0.099	0.099	0.100	0.099	0.100	0.100	0.101	0.101	0.100
TimeInSys, **Y**	0.833	0.924	0.935	0.990	0.996	1.008	1.047	1.077	1.130	1.169

Table 2.9 *Replication Averages for Stratification Variate and Response Variable in PS*

By the Central Limit Theorem, the average service time and time in system becomes normally distributed. Hence, using the fact that each replications consisted of 10,000 entities, we find the theoretical mean and the standard deviation of the average service time as follows:

$$\overline{\mu}_S = \frac{1}{n}\sum_{i=1}^{n}\mu_i = \frac{1}{10000}\sum_{i=1}^{10000}0.1 = 0.1 \text{ and } \overline{\sigma}_S^2 = \frac{1}{n^2}\sum_{i=1}^{n}\sigma_i^2 = \frac{1}{10000^2}\sum_{i=1}^{10000}0.01 = 10^{-6} \Rightarrow \overline{\sigma}_S = 0.01$$

Then, we calculate the boundaries of the strata:

$\zeta_1 = \overline{\mu}_S + \overline{\sigma}_S\phi^{-1}(0.2705) = 0.1 + 0.001\times(-0.6113) = 0.0994$ and

$\zeta_2 = \overline{\mu}_S + \overline{\sigma}_S\phi^{-1}(0.7295) = 0.1 + 0.001\times(+0.6113) = 0.1006$, where ϕ denotes the standard normal distribution.

Hence, we construct three strata as $[-\infty, 0.0994]$, $[0.0994, 0.1006]$, $[0.1006, +\infty]$ and classify the time-in-system values according to the corresponding service time values. After calculating the means and variances of time-in-system values in each stratum, we obtain the Table 2.10:

	Run1	Run2	Run3	Run4	Run5	Run6	Run7	Run8	Run9	Run10
Service Time	0.099	0.099	0.099	0.099	0.100	0.100	0.100	0.100	0.101	0.101
TimeInSys	0.833	0.996	0.924	0.935	0.990	1.008	1.047	1.169	1.077	1.130
Stratum	**1**				**2**				**3**	
Probability	π_1=4/10=0.4				π_2=4/10=0.4				π_3=2/10=0.2	
Average	μ_1=(0.833+0.996+0.924+0.935)/4 =**0.9220**				μ_1=(0.990+1.008+1.047+1.169)/4 =**1.0535**				μ_3=(1.077+1.130)/2 **=1.1035**	
Variance	σ_{1h}^2 =1/3*((0.833-0.922)^2+(0.996-0.922)^2+(0.924-0.922)^2+(0.935-0.922)^2)=**0.0045**				σ_{2h}^2 =1/3*((0.990-1.0535)^2+(1.008-1.0535)^2+(1.047-1.0535)^2+(1.169-1.0535)^2)=**0.0065**				σ_{3h}^2 =(1.077-1.1035)^2+(1.130-1.1035)^2 **=0.0014**	

Table 2.10 *An Example for PS*

Therefore, the post stratified estimator of the time-in-system is

$$\overline{Y}_{ps} = \sum_{h=1}^{L}\pi_h\left(\frac{1}{N_h}\sum_{j=1}^{N_h}Y_{hj}\right) = (0.4\times0.9220 + 0.4\times1.0535 + 0.2\times1.1035) = 1.0109$$

Also, the estimator of the variance of $\overline{Y}_{ps}$ is calculated as

$$Var(\overline{Y}_{ps}) = \sum_{h=1}^{L} \frac{1}{N_h} \times \pi_h^2 \times S_{yh}^2 \Rightarrow$$

$$= \frac{1}{4} \times 0.4^2 \times 0.0045 + \frac{1}{4} \times 0.4^2 \times 0.0065 + \frac{1}{2} \times 0.2^2 \times 0.0014 = 0.000468$$

In order to construct the confidence interval for mean time-in-system, we should determine the effective degrees of freedom, which can be calculated through the following formula as 9.86:

$$\upsilon_e = \frac{\left(\sum_{k=1}^{L} \frac{1}{N_h} \pi_h^2 S_{Yh}^2 \right)^2}{\sum_{k=1}^{L} \frac{\pi_h^4 S_{Yh}^4}{N_h^2 (N_h + 1)}} - 2$$

a 95% confidence interval around the mean is ($t_{9.86,\ 1\text{-}0.025}$=2.233):

$$1.0109 \ \mu \ t_{9.86,1-0.025} \times \sqrt{0.000468} = 1.0109 \ \mu \ 0.0483 = [0.9626;1.10592]$$

As a result, reduction in CI compared to independent sampling case is

$$\frac{0.0717 - 0.0483}{0.0717} = 32.64\%$$

Chapter 3

Analysis of VRTs in the Output Analysis of an M/M/1 Queuing System

Hereafter, we present the experimental results of the variance reduction techniques (VRTs) discussed in the previous chapter: First, we consider the simple M/M/1 system with two traffic rates (ρ): 0.5 and 0.9. Secondly, we analyze a serial line production system consisting of five workstations with limited buffers between these stations and allowing the 'blockage' and 'starvation' of workstations. Finally, we consider an inventory system with the well-known (s,S) policy.

3.1. M/M/1 with ρ=0.9 and ρ=0.5

In this research, all of four techniques, (AV, LHS, CV, and PS) and their combinations are applied to the "time-in-system" variable in M/M/1 queuing system. This system is examined under two different congestion levels or traffic rates: 0.9 and 0.5. The aim of considering two congestion levels is to analyze the behaviour of four methods and their combinations under different utilization and load levels. More specifically, the individual and combined VRTs presented in Table 3.1 are applied for ρ=0.9 and ρ=0.5:

AV	PS (s=2, 3,4,5)
CV	AV+CV
LHS (k=2,3)	LHS (k=2, 3) +CV

Table 3.1 *List of Single and Integrated VRTs that will be Applied to M/M/1*

During the simulation runs, we take the service rate μ as 10 per minute and accordingly, the arrival rate λ as 9 and 5 for two cases, respectively. Furthermore, theoretical values of some statistics for the M/M/1 system are calculated using the Queuing Theory to provide a benchmark for the results obtained in the experiments in Table 3.2:

	ρ=0.9	ρ=0.5
Average Number of Customers in the System	$L = \frac{\lambda}{\mu - \lambda} = \frac{9}{10-9} = 9$	$L = \frac{\lambda}{\mu - \lambda} = \frac{5}{10-5} = 1$
Average Time in the System	$W = \frac{L}{\lambda} = \frac{9}{9} = 1$	$W = \frac{L}{\lambda} = \frac{1}{5} = 0.2$
Average Waiting Time	$W_q = W - W_s = 1 - 0.1 = 0.9$	$W_q = W - W_s = 0.2 - 0.1 = 0.1$

Table 3.2 *Theoretical Values of Queuing Statistics from the Queuing Theory*

We simulate this M/M/1 system for 410,000 entities in total, however, discard the first 10,000 entities as the transient period at both congestion levels. The reason for such a long simulation is to ensure the existence of steady state in the analysis. Even though a much smaller run length is quite enough to estimate a first moment such as average, we need to take longer simulation runs to estimate higher moments such as variance. In each case, we estimate the steady state variance of the system and then calculate the resulting improvement levels. Warm up period is determined using the cumulative average approach. As seen in the Figure 3.1 and 3.2, transient period is longer in highly loaded case, which is around 5,000-6,000. Nevertheless, to be on the conservative side, we take it as 10,000 and collect the statistics for the next 400,000 entities of simulation runs.

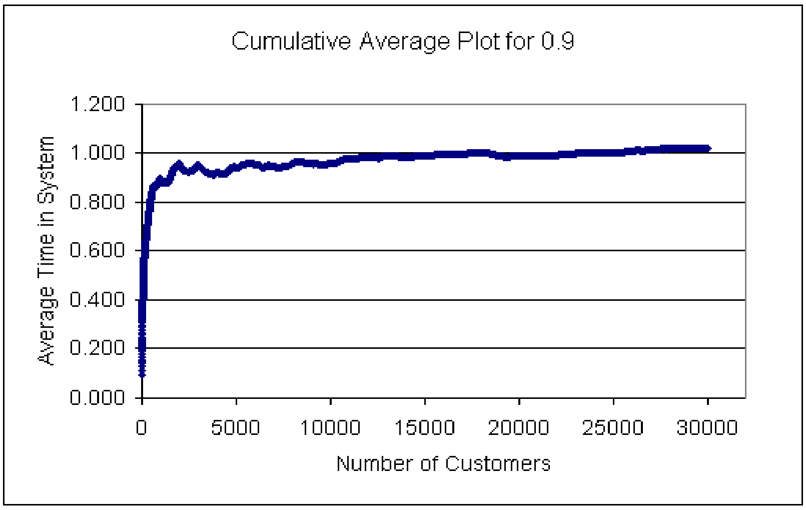

Figure 3.1 *Warm up Period for the 0.9 Utilization Level*

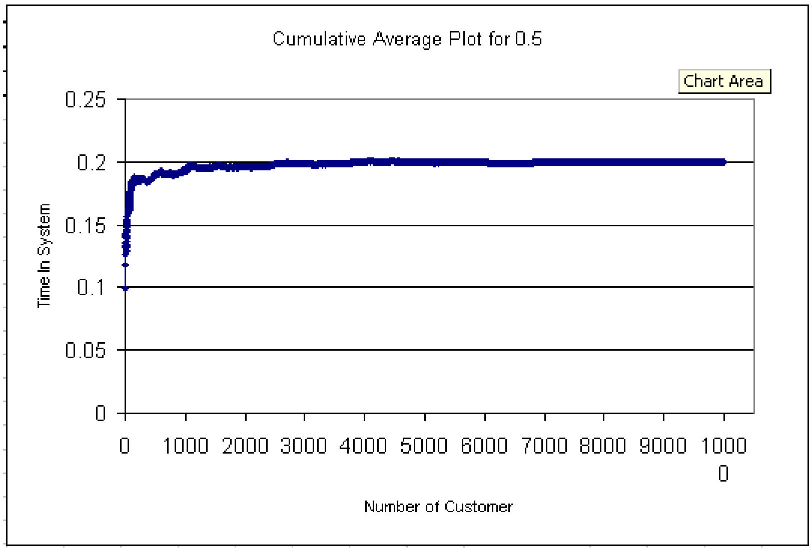

Figure 3.2 *Warm up Period for the 0.5 Utilization Level*

During the simulation experiments, we take 60 replications in all applications of VRTs individually or in combination. In practical cases we are supposed to complete the simulation study in a very limited time and the same replication size considers this assumption. Therefore, a decision should be given whether to apply a VRT with the aim to reduce the variance or not.

3.2. Application of VRTs Individually

3.2.1. Independent Case

Even though we take 60 independent simulation runs to get point and interval estimators on the performance measures, we repeat the experiments 10 times to make sound statistical conclusions. That is, we construct 10 independent confidence intervals. Then the averages of the 10 standard deviation and half-length estimates are used as the benchmark values during the performance evaluation. Improvements are considered in terms of half-length and the standard deviation in this study. Results of 10 experiments for two utilizations with the stream numbers and initial seed values are given in Tables 3.3 and 3.4. “CV” means ‘Coefficient of Variation’ in these tables and throughout this text. “Upper” and “Lower” limits denote the corresponding limits of the confidence interval on the mean.

ρ=0.9								
No	Interarrival Seed	Service Seed	Mean	Std. Dev.	CV	Upper Limit	Lower Limit	Half-length
1	1	3	1.000188	0.004194	0.004193	1.008581	0.991796	**0.008392**
2	1-13867	4	0.998522	0.003396	0.003401	1.005319	0.991726	**0.006796**
3	1-29957	5	0.998968	0.004269	0.004274	1.007510	0.990425	**0.008542**
4	1-30123	6	1.001633	0.003534	0.003528	1.008705	0.994561	**0.007072**
5	1-11979	7	0.996134	0.004292	0.004308	1.004721	0.987546	**0.008588**
6	1-13597	8	1.007590	0.004092	0.004061	1.015778	0.999402	**0.008188**
7	1-10357	3-17291	1.000527	0.003602	0.003600	1.007734	0.993319	**0.007208**
8	1-13565	4-23757	0.999861	0.003586	0.003586	1.007036	0.992685	**0.007175**
9	1-12975	5-29311	0.998864	0.003713	0.003717	1.006293	0.991435	**0.007429**
10	1-28651	6-15457	1.003913	0.003790	0.003775	1.011496	0.996330	**0.007583**

Table 3.3 *Results of the Ten Experiments with the Stream and Initial Seed Numbers for 0.9 Utilization*

ρ=0.5								
No	Interarrival Seed	Service Seed	Mean	Std. Dev.	CV	Upper Limit	Lower Limit	Half-length
1	1	3	0.200025	0.000133	0.000663	0.200291	0.199760	**0.000265**
2	1-13867	4	0.199875	0.000108	0.000538	0.200090	0.199660	**0.000215**
3	1-29957	5	0.199888	0.000134	0.000669	0.200156	0.199621	**0.000267**
4	1-30123	6	0.200089	0.000125	0.000623	0.200339	0.199840	**0.000250**
5	1-11979	7	0.199754	0.000109	0.000545	0.199971	0.199536	**0.000218**
6	1-13597	8	0.200143	0.000124	0.000622	0.200392	0.199894	**0.000249**
7	1-10357	3-17291	0.199998	0.000111	0.000557	0.200221	0.199775	**0.000223**
8	1-13565	4-23757	0.199949	0.000136	0.000680	0.200220	0.199677	**0.000272**
9	1-12975	5-29311	0.200122	0.000112	0.000559	0.200346	0.199898	**0.000224**
10	1-28651	6-15457	0.200220	0.000116	0.000581	0.200453	0.199987	**0.000233**

Table 3.4 *Results of the Ten Experiments with the Stream and Initial Seed Numbers for 0.5 Utilization*

In Tables 3.5 and 3.6, benchmark half-length and standard deviation values are presented.

Estimated Half-length (ρ=0.9)					Estimated Standard Deviation (ρ=0.9)				
Average	Std.Dev.	Upper Limit	Lower Limit	Half-length	Average	Std.Dev.	Upper Limit	Lower Limit	Half-length
0.007697	0.000212	0.007849	0.007546	0.000151	**0.003847**	0.000106	0.003922	0.003771	0.000076

Table 3.5 *Benchmark Half-length and Standard Deviation Values for 0.9 Utilization*

Estimated Half-length (ρ=0.5)					Estimated Standard Deviation (ρ=0.5)				
Average	Std.Dev.	Upper Limit	Lower Limit	Half-length	Average	Std.Dev.	Upper Limit	Lower Limit	Half-length
0.000242	0.000007	0.000247	0.000237	0.000005	**0.000121**	0.000003	0.000123	0.000118	0.000002

Table 3.6 *Benchmark Half-length and Standard Deviation Values for 0.5 Utilization*

According to the CV values in both utilization levels, highly utilized (or highly loaded) systems are more variable than the low utilized systems. Also, the relative precision is very high in both systems. Hence, increasing the utilization of a system increases the variability in the system and thus decreases the predictability of the system performance.

3.2.2. Antithetic Variates (AV)

Since there are two input variables (interarrival and service times) that control the time-in-system in the M/M/1 system, negatively correlated replications are created by inducing negative correlation among both of them or either of them. We firstly apply AV to both input variables simultaneously and then one at a time to interarrival and service times separately. As explained before, 10 different simulation experiments are performed using different random numbers in order to better measure the 10 improvements. The average of these 10 confidence intervals are used to estimate a point and interval estimators on the improvement of AV. We present the stream numbers and initial seed values used in the Table 3.7.

Stream Numbers and Initial Seed Values When AV is applied to:					
1. Interarrival & Service Times		2. Interarrival Times Only		3. Service Times Only	
Interarrival Seed	Service Seed	Interarrival Seed	Service Seed	Interarrival Seed	Service Seed
1	2	1	8,7	1,2	3
3	4	2	8-25557+ 7-30795	1-13867+ 2-21363	4
5	6	3	8-28651+ 7-19655	1-29957+ 2-17239	5
7	8	4	8-10237+ 7-27653	1-30123+ 2-26539	6
1-10235	2-25657	5	8-20721+ 7-23885	1-11979+ 2-10667	7
3-23547	4-26537	6	8-12377+ 7-31267	1-13597+ 2-27893	8
5-21577	6-31257	1-23251	8-13571+ 7-19631	1-10357+ 2-13597	3-17291
7-14259	8-26597	2-17913	8-31235+ 7-12357	1-13565+ 2-27893	4-23757
1-31791	2-13595	3-23157	8-11537+ 7-23591	1-12975+ 2-23597	5-29311
3-13157	4-25793	4-15423	8-11233+ 7-21357	1-28651+ 2-21135	6-15457

Table 3.*7 Stream Numbers and Initial Seed Values for AV*

While applying AV to one input random variable, we use U's in odd numbered replications and (1-U)'s in the even numbered replications as stated before. During this process, we alter the stream number or initial seed value used to generate the other input random variable in order to prevent a positive correlation among the output variable. That is, if we apply AV to interarrival times and leaving service times independent, then, we use different uniform random numbers to generate service times in odd and even numbered replications since we would induce a positive correlation otherwise.

30 complementary pairs of replications are taken in the construction of AV. The negative covariance between the odd and even numbered runs will certainly help us reduce the variation around the mean. Using the variances of odd and even numbered runs and the covariance between them, we calculated the confidence interval and the resulting improvements in the half-lengths. The results are given in Tables 3.8 and 3.9. 'Var(ODD)&Var(EVEN)' shows the variances of odd or even numbered replications among themselves.

AV applied to Interarrival & Service Times (ρ=0.9)									
No	Average	Std. Dev.	Covariance	Correlation	Var(ODD)	Var(EVEN)	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.997036	0.003139	-0.000214	-0.266121	0.000858	0.000752	0.006420	**16.60%**	**18.39%**
2	0.996421	0.002913	-0.000270	-0.346918	0.000745	0.000813	0.005957	**22.61%**	**24.28%**
3	0.999239	0.003025	-0.000382	-0.411237	0.000873	0.000989	0.006187	**19.63%**	**21.36%**
4	0.996928	0.003975	-0.000167	-0.154407	0.000854	0.001377	0.008128	**-5.60%**	**-3.33%**
5	1.003761	0.003278	-0.000517	-0.454861	0.000919	0.001404	0.006703	**12.91%**	**14.79%**
6	0.999061	0.003083	-0.000278	-0.327949	0.000812	0.000884	0.006305	**18.09%**	**19.86%**
7	1.001752	0.003064	-0.000213	-0.276133	0.000857	0.000696	0.006265	**18.61%**	**20.36%**
8	0.998922	0.003501	-0.000064	-0.080252	0.000760	0.000839	0.007160	**6.99%**	**8.99%**
9	0.996018	0.002495	-0.000394	-0.520965	0.000635	0.000899	0.005101	**33.73%**	**35.15%**
10	0.998007	0.003887	-0.000037	-0.039517	0.000841	0.001046	0.007949	**-3.27%**	**-1.04%**

Table 3.8 *Results of AV applied to both Interarrival and Service Times for 0.9 Utilization*

AV applied to Interarrival & Service Times (ρ=0.5)									
No	Average	Std. Dev.	Covariance	Correlation	Var(ODD)	Var(EVEN)	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.200040	0.000088	-0.0000001	-0.146596	0.000000	0.000001	0.000181	**25.26%**	**26.87%**
2	0.199801	0.000099	-0.0000004	-0.382040	0.000001	0.000001	0.000202	**16.37%**	**18.17%**
3	0.200032	0.000096	-0.0000004	-0.428965	0.000001	0.000001	0.000196	**18.90%**	**20.64%**
4	0.199887	0.000106	-0.0000004	-0.365979	0.000001	0.000001	0.000216	**10.62%**	**12.54%**
5	0.200067	0.000107	-0.0000003	-0.332830	0.000001	0.000001	0.000220	**9.15%**	**11.10%**
6	0.200102	0.000099	-0.0000003	-0.371479	0.000001	0.000001	0.000202	**16.26%**	**18.06%**
7	0.200173	0.000108	-0.0000003	-0.279746	0.000001	0.000001	0.000221	**8.40%**	**10.38%**
8	0.200114	0.000093	-0.0000002	-0.272896	0.000001	0.000001	0.000190	**21.40%**	**23.09%**
9	0.199870	0.000089	-0.0000005	-0.486697	0.000001	0.000001	0.000182	**24.55%**	**26.17%**
10	0.199993	0.000114	-0.000000001	-0.013538	0.000000	0.000001	0.000233	**3.56%**	**5.63%**

Table 3.9 *Results of AV applied to both Interarrival and Service Times for 0.5 Utilization*

In Table 3.10, the average improvement levels are given for two utilizations. In this table, the average values are the averages of 'Improv. (Half Length)' columns in tables 3.8 and 3.9.

	AV to Interarrival & Service Times					Average Correlation
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length	
ρ=0.9	**14.03%**	3.76%	22.53%	5.53%	8.50%	-0.287836
ρ=0.5	**15.45%**	2.31%	20.68%	10.22%	5.23%	-0.308077

Table 3.10 *Confidence Intervals for Improvements when AV applied to Interarrival and Service Times*

According to these improvements, AV when applied to both interarrival and service times seems to perform better for the low utilized system. However, the difference between 15.45% and 14.03% is not significant to make such a conclusion. Moreover, half-length estimated for the 0.9 utilization is larger, which means that AV provides the improvement, ranging in a larger spectrum at high utilization level. Hence, AV produces more consistent improvements at low utilization level. In truth, the results obtained in this study comply with the theoretical expectations.

Negative covariance values are sufficient for an improvement by the theorem (Ross[28]). However, we observe two negative improvements at high utilization even though the covariance values are negative. Actually, this theorem is valid in the infinity case and thus no guarantee can be given for a reduction in the half-length or variance in the finite sample.

Another interesting observation is that in the high utilization, variances of odd and even numbered runs do not converge to the same value until four digits and there are considerable differences among those values. The variance in the highly utilized system does not seem to reach to the steady state even in this huge run length of 400,000 entities and thus, much longer runs may be required. This follows from the fact in the following statistics: high moments reach the steady state later than the lower moments. Since variance is a second moment statistic, a longer warm-up period should be taken. In this study, we estimate the average time-in-system, a first moment statistic. Furthermore, in the low utilized (and less variable system), variances are nearly the same in the first six digits. By this observation, we confirm our previous result that the warm-up period is longer in the highly variable system in terms of the same performance measure as compared to the low variable system.

As a result, instead of taking 60 independent runs, inducing negative correlation among the runs improves the precision in high and low utilizations. Observing the relations between the correlations and improvements, there exists an association between them. However, this is not always the case.

The theoretical basis for this difference in the improvements can be deducted from the following formula:

$$Var[\overline{X}(n)] = \frac{Var(X_j)}{n} = \frac{Var(X_j^{(1)}) + Var(X_j^{(2)}) + 2Cov(X_j^{(1)}, X_j^{(2)})}{4n} = \frac{Var(X_j^{(1)})}{2n}(1+\rho)$$

Since odd and even numbered runs are independent on their own, we expect the odd and even numbered pairs have the same variance, which should be asymptotically equal to the variance of the independent case. As a result, a comparison of the improvement in the half-length due to the application of AV with the independent case, we obtain the following result:

$$\frac{E[hl_{AV}]}{E[hl_{IND}]} = \frac{t_{n-1,1-\alpha} \times \sqrt{\dfrac{Var(X_j^{(i)})}{n} \times \dfrac{(1+\rho)}{2}}}{t_{2.n-1,1-\alpha} \times \sqrt{\dfrac{Var(X_j)}{2n}}} = \frac{t_{n-1,1-\alpha}}{t_{2.n-1,1-\alpha}} \times \sqrt{1+\rho}\ ,$$

where we assume $Var(X_j^{(1)}) \approx Var(X_j^{(2)}) \approx Var(X_j)$. Since the only difference between the improvements in the two utilizations is $\sqrt{(1+\rho)}$, the performance of AV directly depends on the magnitude of the induced negative correlation. In the finite sample case, differences among the variance values significantly affect the performance of AV. However, asymptotically, performance of AV depends only on the induced correlation among the pairs when the variances converge to the same value. There is a substantial difference between the variances of odd and even numbered runs and the independent case (i.e, $Var(X_j^{(1)}) \approx Var(X_j^{(2)}) \approx Var(X_j)$ is not true) in our case and there is a non-negligible difference between the t values for 29 and 59 degrees of freedom. Thus, the improvements are resulted from not only the induced correlation but also the differences of variances in odd and even numbered runs and the t values.

Besides, AV has been applied only to the interarrival or service times one at a time. The results of the interarrival case are summarized in Tables 3.11 - 3.13.

AV to Interarrival Times (ρ=0.9)									
No	Average	Std. Dev.	Covariance	Correlation	Var(ODD)	Var(EVEN)	Half-length	Improv. (Half L.)	(Std. D.)
1	0.991032	0.003123	-0.000049	-0.077301	0.000709	0.000558	0.006387	**17.03%**	**18.81%**
2	0.993764	0.003737	-0.000083	-0.092577	0.001145	0.000696	0.007641	**0.73%**	**2.86%**
3	0.999555	0.002653	-0.000199	-0.349388	0.000375	0.000869	0.005426	**29.50%**	**31.02%**
4	1.001923	0.003849	-0.000035	-0.038831	0.001060	0.000788	0.007870	**-2.25%**	**-0.05%**
5	1.002857	0.003325	-0.000239	-0.265441	0.000865	0.000941	0.006800	**11.66%**	**13.56%**
6	1.000112	0.003177	-0.000087	-0.132050	0.000478	0.000908	0.006498	**15.59%**	**17.40%**
7	1.004749	0.003050	-0.000263	-0.322765	0.000920	0.000723	0.006238	**18.96%**	**20.70%**
8	0.998499	0.003516	-0.000127	-0.150661	0.001068	0.000671	0.007190	**6.59%**	**8.60%**
9	0.999564	0.003079	-0.000182	-0.242296	0.000735	0.000766	0.006296	**18.21%**	**19.97%**
10	1.002320	0.003639	-0.000022	-0.027353	0.000957	0.000676	0.007442	**3.32%**	**5.40%**

Table 3.11 *Results of AV applied to Interarrival Times Only for 0.9 Utilization*

AV to Interarrival Times (ρ=0.5)									
No	Average	Std. Dev.	Covariance	Correlation	Var(ODD)	Var(EVEN)	Half-length	Improv. (Half L.)	(Std. D.)
1	0.199953	0.000103	0.000000	-0.181993	0.000001	0.000001	0.000211	**12.49%**	**14.37%**
2	0.199952	0.000119	0.000000	-0.160118	0.000001	0.000001	0.000243	**-0.50%**	**1.66%**
3	0.199955	0.000082	0.000000	-0.339971	0.000001	0.000001	0.000167	**30.73%**	**32.22%**
4	0.200154	0.000126	0.000000	-0.029258	0.000001	0.000001	0.000258	**-6.75%**	**-4.45%**
5	0.200139	0.000118	0.000000	-0.137161	0.000001	0.000001	0.000242	**-0.26%**	**1.89%**
6	0.199954	0.000109	0.000000	0.084864	0.000001	0.000001	0.000222	**7.98%**	**9.96%**
7	0.200043	0.000088	0.000000	-0.315066	0.000001	0.000000	0.000181	**25.12%**	**26.73%**
8	0.200003	0.000114	0.000000	-0.329022	0.000001	0.000001	0.000234	**3.14%**	**5.23%**
9	0.200040	0.000104	0.000000	-0.241860	0.000001	0.000001	0.000213	**11.73%**	**13.63%**
10	0.199998	0.000116	0.000000	-0.005593	0.000001	0.000001	0.000237	**2.09%**	**4.20%**

Table 3.12 *Results of AV applied to Interarrival Times Only for 0.5 Utilization*

	AV to Interarrival Times					
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length	Average Correlation
ρ=0.9	**11.93%**	3.10%	18.95%	4.92%	7.01%	-0.169866
ρ=0.5	**8.58%**	3.74%	17.02%	0.13%	8.45%	-0.165518

Table 3.13 *Confidence Intervals for Improvements when AV applied to Interarrival Times Only*

According to the average improvements, AV when applied to the interarrival times seems to be more effective in the high utilization. In this case, average values of 11.93% and 8.58% have been observed in the high and low utilizations, respectively. As the utilization rate gets lower, time-in-system value becomes nearly equal to the service time and it does not depend on the interarrival times. Thus, interarrival times affect the time-in-system more in the high utilization. Hence, inducing negative correlation among the interarrival times reduces the variance in the time-in-system more in the high utilization or highly variable systems. Comparing the half-lengths, AV produces more consistent improvements in the high utilization also. Nevertheless, one and three negative improvements are observed in the high and low utilizations, which indicate that variance reduction may not be guaranteed in the finite sample case. Then we present the results when AV applied to the service times in Tables 3.14-3.16.

AV to Service Times (ρ=0.9)									
No	Average	Std. Dev.	Covariance	Correlation	Var(ODD)	Var(EVEN)	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.990641	0.004602	0.000108	0.093498	0.001078	0.001247	0.009411	**-22.27%**	**-19.64%**
2	0.998969	0.003774	0.000080	0.102982	0.000751	0.000798	0.007718	**-0.26%**	**1.89%**
3	1.002114	0.003485	-0.000383	-0.344740	0.001114	0.001110	0.007126	**7.42%**	**9.41%**
4	1.002347	0.002513	-0.000531	-0.592550	0.000754	0.001067	0.005139	**33.23%**	**34.67%**
5	0.998400	0.003490	-0.000338	-0.315953	0.001087	0.001050	0.007138	**7.27%**	**9.27%**
6	0.993916	0.003106	-0.000280	-0.327524	0.000944	0.000773	0.006352	**17.48%**	**19.26%**
7	1.002385	0.002704	-0.000231	-0.351281	0.000537	0.000801	0.005530	**28.15%**	**29.70%**
8	1.002067	0.003814	0.000190	0.283984	0.000828	0.000538	0.007799	**-1.32%**	**0.86%**
9	0.995802	0.003650	-0.000064	-0.074041	0.000796	0.000931	0.007465	**3.02%**	**5.11%**
10	0.993682	0.003011	-0.000010	-0.017595	0.000508	0.000599	0.006157	**20.01%**	**21.73%**

Table 3.14 *Results of AV applied to Service Times Only for 0.9 Utilization*

AV to Service Times (ρ=0.5)									
No	Average	Std. Dev.	Covariance	Correlation	Var(ODD)	Var(EVEN)	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.199787	0.000129	0.000000	-0.188066	0.000001	0.000001	0.000264	**-9.29%**	**-6.94%**
2	0.199905	0.000111	0.000000	-0.067266	0.000001	0.000001	0.000227	**5.86%**	**7.89%**
3	0.200008	0.000095	0.000000	-0.429310	0.000001	0.000001	0.000194	**19.74%**	**21.47%**
4	0.199996	0.000120	0.000000	-0.199269	0.000001	0.000001	0.000246	**-1.85%**	**0.34%**
5	0.200040	0.000108	0.000000	-0.296892	0.000001	0.000001	0.000221	**8.43%**	**10.40%**
6	0.199926	0.000117	0.000000	-0.228492	0.000001	0.000001	0.000239	**1.13%**	**3.25%**
7	0.199988	0.000066	0.000000	-0.593388	0.000001	0.000001	0.000136	**43.81%**	**45.02%**
8	0.199991	0.000098	0.000000	-0.242274	0.000001	0.000001	0.000200	**17.18%**	**18.96%**
9	0.199890	0.000094	0.000000	-0.306557	0.000001	0.000001	0.000193	**20.13%**	**21.85%**
10	0.200024	0.000118	0.000000	-0.001487	0.000001	0.000001	0.000241	**0.28%**	**2.42%**

Table 3.15 *Results of AV applied to Service Times Only for 0.5 Utilization*

	AV to Service Times					Average Correlation
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length	
ρ=0.9	**9.27%**	5.12%	20.84%	-2.29%	11.57%	-0.154322
ρ=0.5	**10.54%**	4.82%	21.44%	-0.36%	10.90%	-0.2553

Table 3.16 *Confidence Intervals for Improvements when AV applied to Service Times Only*

Unlikely to the previous case, AV when applied to service times is more effective in the low utilization. Even though the difference may not seem so significant, comparison of the estimated half-lengths and the upper and lower limits of the improvements support the superiority at the low utilization. The reasoning is the same as in the interarrival case. Furthermore, the confidence intervals include the zero, which means that no improvement can be obtained in the end in this case. Also, there are three and two negative improvements in the high and low utilizations, respectively. Even though negative improvements have been also obtained in the previous two cases, those were not so severe as in this case.

We summarize the average improvement levels when AV is applied to different combinations of the input random variables in Table 3.17.

	Utilization	
AV is applied to	**0.5**	**0.9**
1. Interarrival & Service Times	15.45%	14.03%
2. Interarrival Times	8.58%	11.93%
3. Service Times	10.54%	9.27%

Table 3.17 *Overall Results of AV for both Utilization Levels*

In conclusion, the first case where AV is applied to both interarrival and service times is the best for each utilization and the differences are statistically significant. Moreover, inducing negative correlation among the interarrival and service times separately do their best at high and low utilizations, respectively. Thus AV should be applied to the all input variables of which the performance measure is a monotone function, whenever possible. The overall results are illustrated in Figure 3.3.

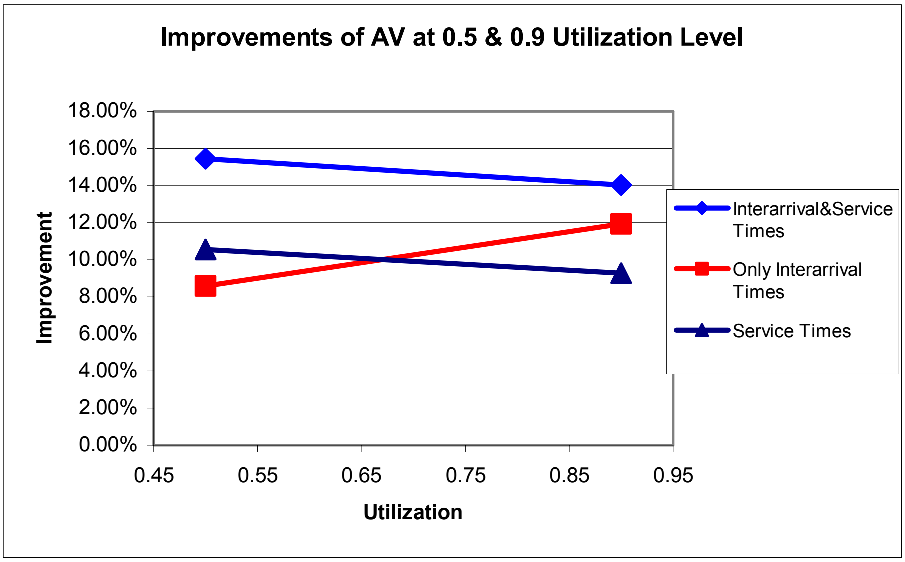

Figure 3.3 *Graph of the Overall Results of AV for both Utilization Levels*

Finally, we will elaborate the monotonicity relationship between the input random variables and the performance measure in the M/M/1 system. The inverse transform method has been used to generate the interarrival and service times. As it is obvious, time-in-system is monotone increasing function of service times. A decrease (increase) in one service time always causes a decrease (increase) in the time-in-system. On the other hand, time-in-system is not a monotonically decreasing function of the interarrival times as opposed to the common sense; it is just a monotone non-increasing function of the interarrival times. More explicitly, increasing or decreasing an interarrival time may not increase or decrease the time-in-system. Even though an increase occurs in the arrival rate, time-in-system may stay the same. Consider an M/M/1 system, for instance. Suppose that the upper diagram Figure 3.4 for the number of customers in the system occurred in one configuration. Increasing second interarrival time, the upper diagram switches to the next. Hence, although the interarrival time increases or interarrival time decreases, no change occurs in the time-in-system value (average of three service times); it stays the same regardless of the increase in the interarrival time. As a result, an increase in the interarrival time does not necessitate an increase in the time-in-system.

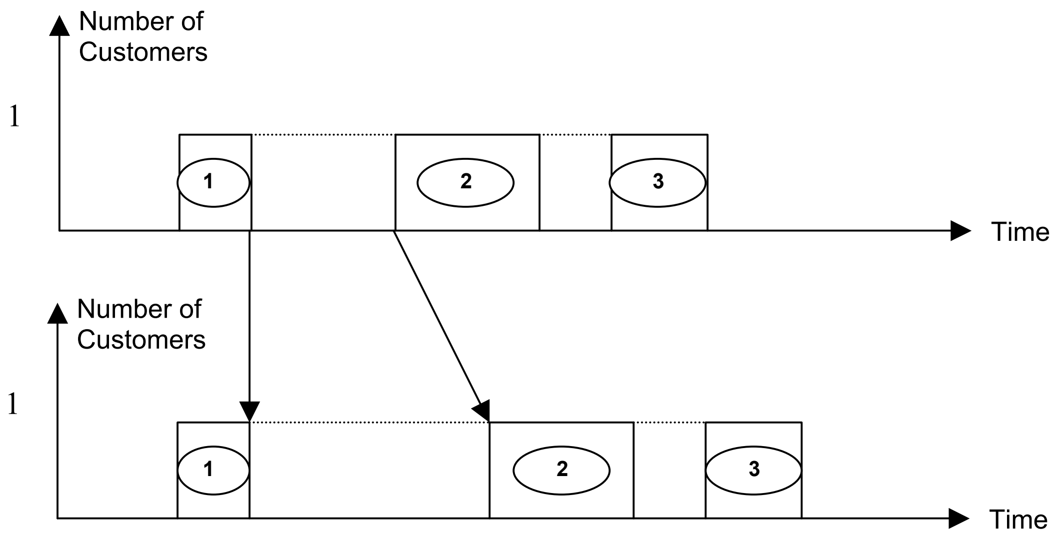

Figure 3.4 *Illustration of the Monotone Relationship between the Interarrival Times and Time-in-system*

3.2.3. Latin Hypercube Sampling (LHS)

Similarly to the AV, LHS is applied to interarrival and service times simultaneously or to either of them separately. Again, 10 sets of experiments are performed using different random numbers in order to get 10 different improvement estimates. In Table 3.18, we present the stream numbers and initial seed values used for LHS. Here, we consider k=2, 3 stratification levels for both the interarrival and the service times. We do not consider the k=4 case since our pilot runs indicated that there did not exist any additional increase. Also, it requires more computational effort.

Stream Numbers and Initial Seed Values When LHS is applied to All Cases			
1. LHS with k=2		2. LHS with k=3	
Interarrival Seed	Service Seed	Interarrival Seed	Service Seed
1,2	1,2	1,2,4	1,2,4
4,5	4,5	5,6,7	5,6,7
6,7	6,7	1-10235+ 2-25657+ 4-26537	1-10235+ 2-25657+ 4-26537
1-10357+ 2-13597	1-10357+ 2-13597	5-21577+ 6-31257+ 7-14259	5-21577+ 6-31257+ 7-14259
4-17291+ 5-12795	4-17291+ 5-12795	1-10357+ 2-13597+ 4-17291	1-10357+ 2-13597+ 4-17291
6-11599+ 7-28711	6-11599+ 7-28711	5-12795+ 6-11599+ 7-28711	5-12795+ 6-11599+ 7-28711
1-13565+ 2-27893	1-13565+ 2-27893	1-13565+ 2-27893+ 4-23547	1-13565+ 2-27893+ 4-23547
4-23547+ 5-10235	4-23547+ 5-10235	5-18315+ 6-25557+ 7-19655	5-10235+ 6-25557+ 7-19655
6-25557+ 7-19655	6-25557+ 7-19655	1-28651+ 2-21135+ 4-15423	1-28651+ 2-21135+ 4-15423
1-28651+ 2-21135	1-28651+ 2-21135	5-29311+ 6-15457+ 7-31791	5-29311+ 6-15457+ 7-31791

Table 3.18 *Stream Numbers and Initial Seed Values for LHS*

As in AV, to apply LHS to one variable, we use different uniform random numbers for the other variable in all micro replications corresponding to one macro replication. The purpose is to prevent the possibility of a positive correlation between the micro replications. We consider 30 and 20 macro replications for k=2 and k=3 stratification levels, respectively. Firstly, we apply LHS to both the interarrival and service times into two (three) levels for k=2 (k=3) simultaneously and then we stratify the interarrival times and service times separately.

LHS with k=2

Firstly, we induce negative correlation among both the interarrival and service times. Results are given in Tables 3.19 and 3.20.

LHS with k=2 applied to Interarrival & Service Times										
	ρ=0.9					**ρ=0.5**				
No	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.993609	0.003152	0.006446	**16.26%**	**18.06%**	0.199854	0.00013	0.000267	**-10.38%**	**-8.01%**
2	0.996855	0.003278	0.006703	**12.92%**	**14.79%**	0.199926	0.000129	0.000264	**-9.32%**	**-6.96%**
3	0.995975	0.003024	0.006184	**19.66%**	**21.39%**	0.199918	0.000095	0.000195	**19.26%**	**21.00%**
4	0.997566	0.003424	0.007002	**9.03%**	**10.99%**	0.199887	0.000124	0.000253	**-4.82%**	**-2.56%**
5	0.996025	0.002454	0.005019	**34.79%**	**36.20%**	0.199795	0.000079	0.000161	**33.44%**	**34.87%**
6	0.995261	0.002722	0.005566	**27.69%**	**29.25%**	0.199886	0.000088	0.00018	**25.35%**	**26.95%**
7	0.995305	0.003151	0.006444	**16.28%**	**18.09%**	0.199982	0.000099	0.000203	**16.11%**	**17.91%**
8	1.002525	0.003613	0.007389	**4.00%**	**6.07%**	0.200044	0.0001	0.000205	**15.26%**	**17.09%**
9	0.998962	0.003851	0.007875	**-2.31%**	**-0.11%**	0.19995	0.000113	0.000232	**3.97%**	**6.04%**
10	0.998266	0.003124	0.006389	**17.00%**	**18.79%**	0.199975	0.000117	0.000238	**1.33%**	**3.45%**

Table 3.19 *Results of LHS with k=2 when applied to both Interarrival and Service Times*

	LHS with k=2 - Interarrival & Service Times						Average Correlation
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length		
ρ=0.9	**15.53%**	3.40%	23.21%	7.85%	7.68%		-0.193034
ρ=0.5	**9.02%**	4.77%	19.79%	-1.76%	10.77%		-0.142275

Table 3.20 *Confidence Intervals for Improvements when LHS with k=2 applied to Both Interarrival and Service Times*

The results indicate that LHS with k=2 yields more improvement in the half-lengths at the high utilization rate (0.9). Upper and lower limits of this interval confirm this observation. Even though the 95% confidence interval at the high utilization does not include the zero and in fact it has a significant lower bound (7.85%), this is not the case for the low utilization. Therefore, LHS with k=2 when applied to both interarrival and service times performs better in the high utilization.

Among 10 different experiments (and corresponding improvement values), we see only one occasion in which LHS degrade the precision (-2.31%), which is actually not significant. On the other hand, there exist three negative improvements and two of them can be considered significant. Once again this indicates that there is no guarantee to reduce the variance by LHS in a finite sample space. Theoretically, LHS should provide a positive improvement as long as the performance measure is a monotone function of the input variables among which the negative correlation is induced. We present the results of the service time case in Tables 3.21 and 3.22.

LHS with k=2 applied to Service Times Only										
	ρ=0.9					**ρ=0.5**				
No	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	1.008279	0.00401	0.0082	**-6.53%**	**-4.24%**	0.20019	0.000097	0.000199	**17.81%**	**19.58%**
2	0.99422	0.002964	0.006061	**21.26%**	**22.95%**	0.199964	0.00011	0.000224	**7.13%**	**9.13%**
3	1.000494	0.003809	0.00779	**-1.21%**	**0.97%**	0.199894	0.000118	0.00024	**0.54%**	**2.68%**
4	1.001589	0.002811	0.005749	**25.31%**	**26.91%**	0.199949	0.000099	0.000202	**16.41%**	**18.21%**
5	0.99518	0.00414	0.008467	**-10.00%**	**-7.63%**	0.199998	0.00012	0.000245	**-1.57%**	**0.62%**
6	0.993925	0.003349	0.006849	**11.02%**	**12.94%**	0.199899	0.000127	0.000259	**-7.09%**	**-4.79%**
7	0.999201	0.003537	0.007233	**6.04%**	**8.06%**	0.200113	0.000108	0.000222	**8.30%**	**10.27%**
8	1.001207	0.003842	0.007856	**-2.07%**	**0.13%**	0.200058	0.000143	0.000292	**-21.03%**	**-18.43%**
9	0.997564	0.003854	0.007881	**-2.39%**	**-0.19%**	0.200002	0.000149	0.000305	**-26.05%**	**-23.34%**
10	1.006111	0.004238	0.008668	**-12.61%**	**-10.18%**	0.200165	0.00013	0.000266	**-10.13%**	**-7.76%**

Table 3.21 *Results of LHS with k=2 when applied to Service Times Only*

	LHS with k=2 to Service Times						Average Correlation
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length		
ρ=0.9	**2.88%**	4.06%	12.05%	-6.28%	9.17%		-0.129162
ρ=0.5	**-1.57%**	4.66%	8.97%	-12.11%	10.54%		-0.120402

Table 3.22 *Confidence Intervals for Improvements when LHS with k=2 applied to Service Times Only*

Since both confidence intervals include zero and the averages are in fact very close to zero, we cannot say that LHS with k=2 when applied to service times will produce a smaller confidence interval. The probability of success is much smaller in this case. Therefore, LHS with k=2 should not be applied to only service times in the M/M/1 system in order to reduce variation, especially at the low utilizations.

LHS with k=2 applied to Interarrival Times Only										
	ρ=0.9					ρ=0.5				
No	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.994231	0.003666	0.007497	**2.60%**	**4.69%**	0.199868	0.000107	0.000219	**9.47%**	**11.41%**
2	0.997327	0.003706	0.007579	**1.54%**	**3.66%**	0.20005	0.00013	0.000265	**-9.67%**	**-7.31%**
3	1.0022	0.003298	0.006745	**12.38%**	**14.26%**	0.200033	0.000125	0.000256	**-5.87%**	**-3.59%**
4	1.00323	0.00414	0.008466	**-9.99%**	**-7.62%**	0.199933	0.000124	0.000254	**-5.11%**	**-2.84%**
5	1.000784	0.002547	0.005209	**32.32%**	**33.78%**	0.200014	0.000115	0.000235	**2.85%**	**4.94%**
6	1.003562	0.00447	0.009141	**-18.76%**	**-16.20%**	0.200107	0.000125	0.000255	**-5.40%**	**-3.13%**
7	0.998423	0.003827	0.007826	**-1.68%**	**0.51%**	0.200036	0.000102	0.000209	**13.48%**	**15.34%**
8	1.001035	0.004638	0.009485	**-23.22%**	**-20.57%**	0.199793	0.000152	0.00031	**-28.45%**	**-25.68%**
9	1.000419	0.003241	0.006627	**13.90%**	**15.76%**	0.200036	0.000119	0.000243	**-0.54%**	**1.62%**
10	1.005825	0.003903	0.007982	**-3.70%**	**-1.47%**	0.199989	0.000126	0.000258	**-6.68%**	**-4.39%**

Table 3.23 *Results of LHS with k=2 when applied to Interarrival Times Only*

	LHS with k=2 to Interarrival Times					
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length	Average Correlation
ρ=0.9	**0.54%**	5.17%	12.22%	-11.14%	11.68%	0.008936
ρ=0.5	**-3.59%**	3.63%	4.62%	-11.80%	8.21%	-0.007998

Table 3.24 *Confidence Intervals for Improvements when LHS with k=2 applied to Interarrival Times only*

Lastly, we present the results of the interarrival time case in Tables 3.23 and 3.24. Similarly to the previous case, both confidence intervals include the zero and estimated averages are very close to zero. This means that applying LHS with k=2 to interarrival times is not quite effective in reducing the variance. On the other hand, by examining the average improvement values at two utilizations, LHS with k=2 seems to perform better at the high utilizations when LHS is applied to interarrival times.

In conclusion, variance reduction cannot be observed all the time even if it is expected theoretically. Also, LHS should be applied to all input variables whenever possible in order to increase the gain similarly to AV.

LHS with k=3

In Tables 3.25 and 3.26, we present the results when LHS with k=3 is applied to both interarrival and service times.

LHS with k=3 applied to Interarrival & Service Times										
	ρ=0.9					ρ=0.5				
No	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.995916	0.002596	0.005426	**29.51%**	**32.51%**	0.199857	0.000094	0.000196	**19.04%**	**22.49%**
2	1.00187	0.003931	0.008217	**-6.75%**	**-2.20%**	0.199932	0.000122	0.000254	**-5.13%**	**-0.66%**
3	0.99783	0.003297	0.006892	**10.47%**	**14.28%**	0.200043	0.000087	0.000182	**24.63%**	**27.84%**
4	1.00192	0.003072	0.00642	**16.59%**	**20.14%**	0.200041	0.000087	0.000181	**25.10%**	**28.29%**
5	0.997187	0.003818	0.00798	**-3.68%**	**0.74%**	0.199951	0.000116	0.000242	**-0.36%**	**3.92%**
6	1.003304	0.00221	0.004619	**39.99%**	**42.55%**	0.199995	0.000141	0.000295	**-21.92%**	**-16.73%**
7	1.005932	0.002378	0.00497	**35.43%**	**38.18%**	0.200068	0.000104	0.000217	**10.18%**	**14.01%**
8	0.99972	0.003407	0.00712	**7.50%**	**11.44%**	0.199812	0.000108	0.000225	**6.98%**	**10.94%**
9	0.999174	0.003594	0.007511	**2.42%**	**6.57%**	0.199923	0.000104	0.000216	**10.44%**	**14.25%**
10	0.99775	0.002871	0.005999	**22.06%**	**25.38%**	0.199945	0.000099	0.000208	**14.10%**	**17.75%**

***Table 3.25** Results of LHS with k=3 when applied to both Interarrival and Service Times*

	LHS with k=3 - Interarrival & Service Times				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
ρ=0.9	**15.35%**	5.12%	26.93%	3.78%	11.57%
ρ=0.5	**8.30%**	4.57%	18.62%	-2.01%	10.32%

***Table 3.26** Confidence Intervals for Improvements when LHS with k=3 applied to Both Interarrival and Service Times*

The results indicate that LHS with k=3 certainly performs better in the high utilization. The confidence interval does not include the zero, this means that LHS with k=3 provides substantial improvements in the precision. However, it does not always guarantee such an improvement. As seen in Table 3.25, there are two negative values in the half-lengths and one in the standard deviation at the high utilization. Similarly, there exist two significant negative improvements at the low utilization. This suggests that the success of the LHS with k=3 is less likely in the low utilization system.

After that, we present the results of the service time case in Tables 3.27 and 3.28. In general, LHS with k=3 when applied to service times is more effective at the low utilization. This is due to the large proportion of the service times in the time-in-system in the low utilization. For 0.5 utilization, service times constitute the 50% of the time-in-system while in the 0.9 utilization, it constitutes only 10% of the time-in-system, theoretically. This should be the reason for the difference in the average improvements. On the other hand, both confidence intervals include the zero, therefore, LHS with k=3 may not provide an improvement in both utilizations. Looking at the individual improvement levels at both utilizations, LHS with k=3 provided improvement 90% and 60% of the time at the low and high utilizations, respectively.

LHS with k=3 applied to Service Times Only										
	ρ=0.9					ρ=0.5				
No	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	1.003915	0.003284	0.006864	**10.83%**	**14.62%**	0.200070	0.000106	0.000222	**8.02%**	**11.94%**
2	0.996763	0.003895	0.008141	**-5.76%**	**-1.26%**	0.199902	0.000099	0.000206	**14.63%**	**18.26%**
3	0.996664	0.004057	0.008480	**-10.16%**	**-5.47%**	0.199928	0.000110	0.000229	**5.10%**	**9.14%**
4	1.006282	0.003831	0.008007	**-4.03%**	**0.40%**	0.200110	0.000101	0.000210	**13.03%**	**16.73%**
5	1.001177	0.003647	0.007622	**0.98%**	**5.20%**	0.200053	0.000108	0.000225	**6.97%**	**10.94%**
6	1.001091	0.003566	0.007454	**3.16%**	**7.29%**	0.199996	0.000109	0.000227	**6.10%**	**10.10%**
7	0.998712	0.003295	0.006887	**10.52%**	**14.33%**	0.200051	0.000109	0.000227	**5.92%**	**9.93%**
8	1.004098	0.002890	0.006041	**21.52%**	**24.86%**	0.200030	0.000114	0.000239	**1.06%**	**5.27%**
9	1.000270	0.002632	0.005501	**28.54%**	**31.58%**	0.199944	0.000096	0.000201	**16.90%**	**20.44%**
10	1.000082	0.004845	0.010127	**-31.56%**	**-25.96%**	0.199870	0.000141	0.000296	**-22.33%**	**-17.12%**

Table 3.27 *Results of LHS with k=3 when applied to Service Times Only*

	LHS with k=3 to Service Times				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
ρ=0.9	**2.40%**	5.38%	14.57%	-9.76%	12.16%
ρ=0.5	**5.54%**	3.45%	13.34%	-2.26%	7.80%

Table 3.28 *Confidence Intervals for Improvements when LHS with k=3 applied to Service Times Only*

Lastly, we present the results of the intearrival time case in Tables 3.29 and 3.30.

LHS with k=3 applied to Interarrival Times Only										
	ρ=0.9					ρ=0.5				
No	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.999298	0.002806	0.005865	**23.81%**	**27.05%**	0.199966	0.000132	0.0002759	**-14.21%**	**-9.34%**
2	0.999922	0.004109	0.0085873	**-11.56%**	**-6.81%**	0.199985	0.000129	0.0002695	**-11.55%**	**-6.80%**
3	0.998163	0.002665	0.0055696	**27.64%**	**30.72%**	0.199995	0.000111	0.0002319	**3.99%**	**8.08%**
4	1.003785	0.003863	0.0080736	**-4.89%**	**-0.42%**	0.199944	0.000135	0.000282	**-16.73%**	**-11.76%**
5	0.993104	0.003003	0.0062772	**18.45%**	**21.92%**	0.199829	0.000105	0.0002194	**9.20%**	**13.06%**
6	1.005957	0.003961	0.008278	**-7.54%**	**-2.96%**	0.200093	0.000126	0.000264	**-9.26%**	**-4.60%**
7	0.99662	0.002807	0.0058662	**23.79%**	**27.03%**	0.199954	8.32E-05	0.0001739	**28.01%**	**31.08%**
8	1.000798	0.003552	0.0074231	**3.56%**	**7.67%**	0.200166	0.000126	0.0002627	**-8.73%**	**-4.10%**
9	1.003869	0.003453	0.0072178	**6.23%**	**10.22%**	0.199987	8.68E-05	0.0001814	**24.93%**	**28.13%**
10	0.996839	0.003212	0.0067138	**12.78%**	**16.49%**	0.199908	8.97E-05	0.0001875	**22.39%**	**25.70%**

Table 3.29 *Results of LHS with k=3 when applied to Interarrival Times Only*

	LHS with k=3 to Interarrival Times				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
ρ=0.9	**9.23%**	4.49%	19.38%	-0.92%	10.15%
ρ=0.5	**2.81%**	5.48%	15.20%	-9.59%	12.39%

Table 3.30 *Confidence Intervals for Improvements when LHS with k=3 applied to Interarrival Times only*

Based on the reasoning in the 'service times' case, interarrival times have more impact on the time-in-system at the high utilization. Thus the superiority of LHS with k=3 at the high utilization should be expected. The lower limit of the confidence interval for the high utilization is negative but very close to zero. Therefore, LHS with k=3 is expected to provide substantial improvements at the high utilization. In Table 3.31, we summarize the results of LHS with stratification levels k=2 and k=3. We plot the graph in Figure 3.5 to explain the remarks more clearly. However, since the variability of the individual improvement levels is very large in each case, we cannot make strong conclusions about these results. This means that the expected average improvement values are not so reliable and have a high variation.

LHS applied to		**0.5**	**0.9**
1. Interarrival & Service Times	k=2	9.02%	15.53%
	k=3	8.30%	15.35%
2. Service Times	k=2	-1.57%	2.88%
	k=3	5.54%	2.40%
3. Interarrival Times	k=2	-3.59%	0.54%
	k=3	2.81%	9.23%

Table 3.31 *Overall Results of LHS for both Utilization Levels*

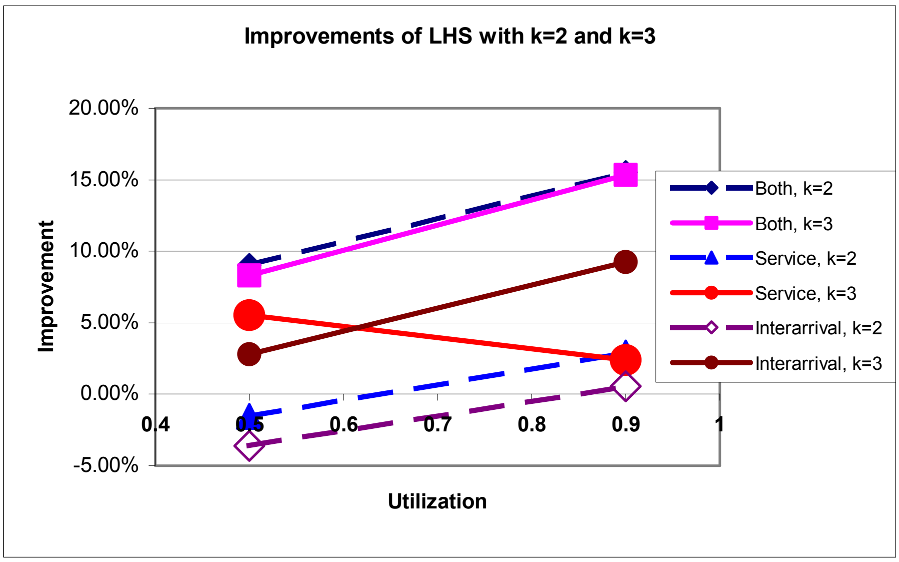

Figure 3.5 *Graph of the Overall Results of LHS for both Utilization Levels*

In conclusion, when LHS is applied to interarrival and service times simultaneously, more improvements are obtained for the k=2 at both utilization levels, but the difference among the improvements are not significant. Thus there is no benefit of increasing the stratification level. On the other hand, when LHS is applied to input random variables individually, increasing the stratification contributes much to the resulting improvement except one case. This exception occurs when the LHS is applied to service times only at the high utilization case. The reason is the small contribution of the service times to the time-in-system at the high utilization. With the same reasoning, increasing the stratification level when LHS is applied to interarrival times causes significant improvements at the high utilization.

Additionally, in order to compare the VRTs, which modify the input random variables to reduce the variance, we combine the results of LHS and AV in Table 3.32 and discuss the inferences subsequently:

	0.5			**0.9**		
	LHS		AV	LHS		AV
	k=2	k=3		k=2	k=3	
1. Interarrival & Service Times	9.02%	8.30%	15.45%	15.53%	15.35%	14.03%
2. Service Times	-1.57%	5.54%	10.54%	2.88%	2.40%	9.27%
3. Interarrival Times	-3.59%	2.81%	8.58%	0.54%	9.23%	11.93%

Table 3.32 *Overall Results of AV and LHS for both Utilization Levels*

The results indicate that AV performs better than LHS and the disparities between the improvements are more substantial when both methods are applied to input random variables individually. One exception occurs when two methods are applied to both input random variables at the high utilization. In this case, LHS provide more improvement than AV while the differences may be neglected. As a result, among the methods modifying the input random variables, AV should be preferred over LHS and it should be applied to all input random variables simultaneously.

3.2.4. Control Variates (CV)

We have chosen 'service time', of which we know the theoretical or expected mean, as our control variate to calculate the modified time-in-system values. We use the 'theoretically known mean' of service time as 0.1 minute for two utilizations. Generally speaking, CV is proposed for systems exhibiting a strong linear association between the control variate and the response variable. Prior to our simulation we aware of the linear dependence between the time-in-system and the service time. Since CV does not require any changes from the input part of the simulation, the data obtained in the independent runs are used in the analysis. The results are shown in Tables 3.33 - 3.35.

No	Control Variates (ρ=0.9)						
	Mean	Std. Dev.	Correlation	a*	Half-length	Improv. (Half L.)	Improv. (Std.D.)
1	0.997732	0.003384	0.590778	114.25197	0.006774	11.99%	**12.03%**
2	0.998301	0.003057	0.435835	82.98352	0.006120	20.49%	**20.53%**
3	1.001203	0.003386	0.609003	107.38617	0.006779	11.93%	**11.97%**
4	1.000717	0.002854	0.589662	98.15910	0.005714	25.76%	**25.80%**
5	1.000999	0.003668	0.519308	133.41193	0.007343	4.61%	**4.66%**
6	1.008686	0.003371	0.566774	119.61404	0.006749	12.32%	**12.36%**
7	1.002815	0.002951	0.573421	108.44179	0.005908	23.25%	**23.29%**
8	1.000122	0.003316	0.380636	61.52116	0.006639	13.75%	**13.80%**
9	0.996070	0.002993	0.591837	109.30697	0.005991	22.16%	**22.20%**
10	0.998788	0.002953	0.626557	103.36989	0.005913	23.18%	**23.22%**

(Average of a*: 103.8447)

***Table 3.33** Results of CV for 0.9 Utilization Level*

No	Control Variates (ρ=0.5)						
	Mean	Std. Dev.	Correlation	a*	Half-length	Improv. (Half L.)	Improv. (Std.D.)
1	0.199929	0.000091	0.728307	4.45402	0.000182	24.68%	**24.72%**
2	0.199864	0.000079	0.681164	4.10910	0.000158	34.74%	**34.77%**
3	0.199977	0.000085	0.773957	4.27306	0.000169	29.87%	**29.91%**
4	0.200050	0.000086	0.722628	4.24419	0.000173	28.54%	**28.58%**
5	0.199943	0.000066	0.795299	5.18007	0.000132	45.32%	**45.35%**
6	0.200184	0.000090	0.686722	4.40490	0.000181	25.07%	**25.11%**
7	0.200090	0.000074	0.748101	4.38010	0.000148	38.71%	**38.74%**
8	0.199968	0.000092	0.733266	4.48986	0.000185	23.42%	**23.46%**
9	0.200020	0.000078	0.716421	3.98283	0.000156	35.38%	**35.41%**
10	0.200037	0.000080	0.729068	3.69448	0.000159	33.99%	**34.02%**

(Average of a*: 4.3213)

***Table 3.34** Results of CV for 0.5 Utilization Level*

	CV				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
ρ=0.9	**16.95%**	2.18%	21.88%	12.01%	4.94%
ρ=0.5	**31.97%**	2.20%	36.95%	26.99%	4.98%

***Table 3.35** Confidence Intervals for Improvements by CV*

Firstly, no negative improvement is observed for both utilization rates unlikely to the AV and LHS. Secondly, service time as a control variate provides significant improvements at two utilizations. Also, the half-lengths constructed for the average improvements are so small CV can provide consistent improvements. In addition, lower limits at both utilizations are much greater than zero. Therefore, CV is a very effective method in reducing the variance and the probability of success is very high. Of course, the improvement depends on the correlation between the selected control variate and the desired performance measure. The correlation is larger when the utilization is smaller if the service time is the control variate.

This can be explained through the Queuing Theory. Time-in-system is the sum of two components: Service time and waiting time in queue. Service time constitutes only 10 % of the time-in-system in highly loaded system while it is the half of the time-in-system in less loaded system. Therefore, service times tell more about the time-in-system in less loaded system. This is the reason for more correlation in low utilized system for the same control variate.

On the other hand, if the waiting time in queue had been chosen as the control variate assuming that we know its theoretical mean, we would have obtained more improvement in the highly loaded system. This is based on the same reasoning: Waiting time in queue constitutes 90% of the time-in-system in highly loaded system while it is only 50% in less loaded system. Thus it tells more about the fluctuations in the time-in-system highly loaded system and we make better adjustments with more confidence in the time-in-system. As a result, increasing the correlation between the control variate and the response variable or choosing a control variate that is highly correlated with the response variable increases the precision of the estimator.

Another observation is that the control coefficient, a^*, is larger at 0.9 utilization. This is entirely due to the scale differences between the control variate and the performance measure. In order to convert any fluctuation of service time from its known mean to the adjustment in the time-in-system, we have to multiply this fluctuation with a larger value in highly loaded system since service time constitutes only the 10% of the time-in-system as stated before.

A comparison of the CV with the previously discussed methods, AV and LHS, reveals that CV causes more improvement in the half-lengths and standard deviation at both utilizations. Therefore, instead of inducing a negative correlation among the replications, applying CV and modifying the output data would be much better.

3.2.5. Poststratified Sampling (PS) (s=2,3,4,5)

In PS, we use the auxiliary information like CV. Even though CV requires a control variate which has a known mean prior to the simulation, PS requires a stratification variate whose distribution is known. We choose the average service time as the stratification variate since by the Central Limit Theorem, it is normally distributed. Mean of the normal distribution is equal to the theoretical mean of the service time and its variance is equal to the theoretical variance of the service time divided by the number of observations in each run, i.e. 400,000 in this case. In this study, we consider four different stratification levels s=2,3,4,5. Again, 10 different simulation experiments are performed using different random numbers and 10 different improvement levels are obtained to construct a half-length for the average improvement by PS.

In order to construct each stratum, we use two possible ways. Firstly, we form intervals of equal probability. Second way has been proposed by Sethi [33], which is the optimal allocation scheme when the stratification variate shows a normal distribution pattern. When long runs of simulation are taken, all random performance measures can be approximated by a normal distribution due to the central limit theorem. Thus, this optimal allocation scheme is expected to be effective in long runs and thus we present the results according to this scheme in this report. Due to space limitations, we exhibit only the improvement amounts in this text.

We present the results of the optimal allocation and the equal probability schemes for two utilization levels corresponding to each stratification level 's' in Tables 3.36 and 3.37. Selecting service time as the stratification variate produces improvements ranging in 18.85-26.21% and 10.80-14.00% in low and high utilizations respectively. As the level of stratification increases, the resulting improvement tends to increase. Nevertheless, as the individual results indicate, this observation may not hold true all the time. Sometimes, increasing 's' decreases the resulting improvement. Furthermore, service time as the stratification variate performs much better in less loaded system with respect to high loaded.

This can be explained using the same reasoning in the CV. In the less utilized systems, any fluctuation in the service time tells more about the true value of time-in-system. In another perspective, during the experimentation, we keep the variability of the service time constant since we just alter the interarrival time to change the congestion level. Hence, increasing rate and thus increasing variability of the interarrival times contributes only to the variability of the waiting time in queue. As a result, as we increase the utilization rate, main decision center shifts from service time to the waiting time. Intuitively, in a lower utilization such as 0.2, we would expect the service time to perform better as the stratification variate, for instance.

Improvement Levels obtained with Optimal Allocation Scheme

No	ρ=0.5				No	ρ=0.9			
	k=2	k=3	k=4	k=5		k=2	k=3	k=4	k=5
1	11.06%	14.31%	22.91%	17.56%	**1**	7.21%	11.10%	10.83%	8.25%
2	27.30%	30.13%	28.52%	29.21%	**2**	13.41%	16.45%	16.66%	18.47%
3	3.52%	19.73%	13.38%	20.32%	**3**	2.96%	2.51%	3.32%	4.23%
4	18.08%	20.55%	20.65%	24.64%	**4**	23.64%	23.45%	24.67%	26.66%
5	32.96%	32.97%	42.23%	37.84%	**5**	0.38%	-4.83%	4.29%	0.51%
6	11.05%	11.94%	24.42%	21.72%	**6**	2.32%	-0.77%	7.67%	8.20%
7	24.52%	33.55%	31.93%	34.38%	**7**	13.09%	12.60%	18.30%	18.46%
8	13.60%	11.73%	14.93%	19.81%	**8**	7.46%	11.63%	11.98%	13.88%
9	22.85%	25.23%	32.72%	30.65%	**9**	24.16%	17.43%	23.84%	24.74%
10	23.55%	25.55%	28.57%	26.00%	**10**	13.97%	18.42%	18.46%	14.91%

Confidence Intervals for Different Stratification Levels	Average	**18.85%**	**22.57%**	**26.02%**	**26.21%**	**10.86%**	**10.80%**	**14.00%**	**13.83%**
	Std. Dev.	2.84%	2.61%	2.75%	2.12%	2.65%	2.88%	2.40%	2.71%
	Upper Limit	25.27%	28.46%	32.23%	31.00%	16.84%	17.30%	19.43%	19.95%
	Lower Limit	12.43%	16.67%	19.82%	21.43%	4.87%	4.30%	8.58%	7.71%
	Half Length	6.42%	5.90%	6.21%	4.79%	5.99%	6.50%	5.43%	6.12%

Table 3.36 *Results of PS and Confidence Intervals for Improvements at each Stratification Level obtained with Optimal Allocation Scheme*

Improvement Levels obtained with Equal Probability Scheme									
No	ρ=0.5				No	ρ=0.9			
	k=2	k=3	k=4	k=5		k=2	k=3	k=4	k=5
1	11.06%	8.46%	17.51%	16.78%	**1**	7.21%	4.52%	9.14%	8.81%
2	27.30%	27.80%	31.23%	28.21%	**2**	13.41%	14.59%	17.77%	14.68%
3	3.52%	17.69%	19.42%	20.74%	**3**	2.96%	3.03%	4.01%	5.43%
4	18.08%	19.41%	24.06%	20.80%	**4**	23.64%	25.91%	25.08%	23.12%
5	32.96%	35.43%	38.25%	43.09%	**5**	0.38%	-2.33%	-0.93%	3.30%
6	11.05%	13.39%	18.77%	21.79%	**6**	2.32%	6.25%	2.89%	8.17%
7	24.52%	28.45%	31.81%	37.52%	**7**	13.09%	12.51%	12.82%	19.93%
8	13.60%	14.85%	15.92%	18.26%	**8**	7.46%	9.38%	12.81%	13.81%
9	22.85%	24.40%	26.82%	31.64%	**9**	24.16%	25.47%	23.69%	23.61%
10	23.55%	23.72%	27.08%	24.97%	**10**	13.97%	14.33%	18.65%	17.25%

Confidence Intervals for Different Stratification Levels		ρ=0.5 k=2	k=3	k=4	k=5	ρ=0.9 k=2	k=3	k=4	k=5
	Average	**18.85%**	**21.36%**	**25.09%**	**26.38%**	**10.86%**	**11.37%**	**12.59%**	**13.81%**
	Std. Dev.	2.84%	2.57%	2.30%	2.75%	2.65%	2.92%	2.80%	2.29%
	Upper Limit	25.27%	27.17%	30.29%	32.58%	16.84%	17.96%	18.92%	18.98%
	Lower Limit	12.43%	15.55%	19.88%	20.17%	4.87%	4.78%	6.26%	8.65%
	Half Length	6.42%	5.81%	5.21%	6.21%	5.99%	6.59%	6.33%	5.17%

Table 3.37 *Results of PS and Confidence Intervals for Improvements at each Stratification Level obtained with Equal Probability Scheme*

Moreover, we observe two negative improvements in the fifth and sixth streams at the high utilization. Thus the application of the PS sometimes may not cause an improvement in the variance even though this probability seems to be very small. CV and PS do not require any change in the simulation run. Hence, we can apply both CV and PS and arrive at the smallest possible confidence interval attainable with these methods. In this case, PS with s=5 provides the best improvement in the low utilization while s=4 achieves the best in the high utilization. Thus s=4 and s=5 are the recommended stratification levels in PS. Also, PS can be applied to the non linear dependency cases. This is indeed another advantage of PS over CV.

In conclusion, increasing the level of stratification tends to increase the improvement. Moreover, when a stratification variate indicating the distribution of a normal distribution is selected, optimal allocation scheme (Sethi [33]) gives very good results. Furthermore, choosing a stratification variate highly correlated with the response variable contributes more to the improvement. Since the application of PS does not consume so much time, the output data can be analyzed under different stratification levels and the best result can be selected. PS does not seem to be a robust VRT; the reported improvements at different utilizations are not very close. This is indicated more clearly in Figure 3.6, which takes 'service time' as the stratification variate. However, this observation totally depends on the relationship between the stratification variate and the response variable. In another case, it may seem to be a very robust technique. Construction of the strata is an important factor in the resulted improvement level as much as the selection of the stratification variate. Essentially, the vital part of PS is the selection of a stratification variate that has a strong correlation with the response variable.

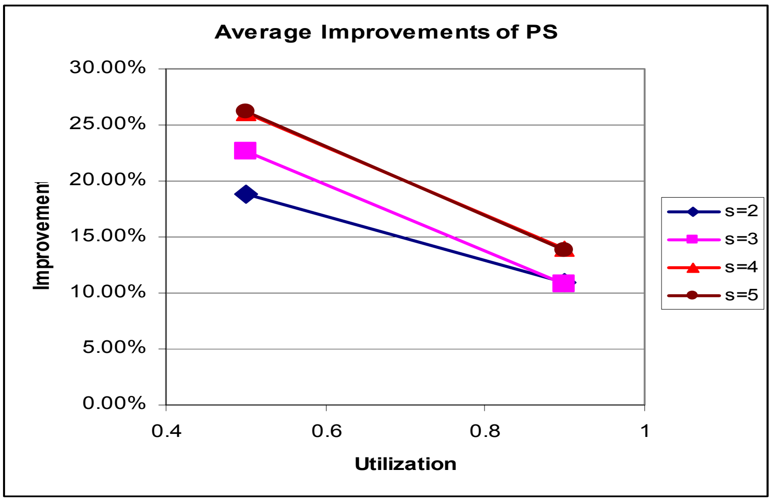

Figure 3.6 *Overall Results of PS at each level when Service Time is the Stratification Variate*

3.2.6. Overview of the Results

We summarize the overall results in Tables 3.38 and 3.39. These are listed as follows:

		0.9					
		Half-length			Standard Deviation		
Method		Improv. (Half L.)	Upper Limit	Lower Limit	Improv. (Std. D.)	Upper Limit	Lower Limit
AV		14.03%	22.53%	5.53%	15.88%	24.19%	7.56%
LHS, k=2		15.53%	23.21%	7.85%	17.35%	24.87%	9.84%
LHS, k=3		15.35%	26.93%	3.78%	17.29%	30.17%	4.41%
CV		16.95%	21.88%	12.01%	16.99%	21.92%	12.05%
PS	s=2	10.86%	16.84%	4.87%	10.95%	16.97%	4.93%
	s=3	10.80%	17.30%	4.30%	10.93%	17.44%	4.41%
	s=4	14.00%	19.43%	8.58%	14.20%	19.63%	8.77%
	s=5	13.83%	19.95%	7.71%	14.03%	20.13%	7.94%

Table 3.38 *Overall Results in Half-length and Standard Deviation by Each Technique for 0.9 Utilization*

		0.5					
		Half-length			Standard Deviation		
Method		Improv. (Half L.)	Upper Limit	Lower Limit	Improv. (Std. D.)	Upper Limit	Lower Limit
AV		15.45%	20.68%	10.22%	17.26%	22.38%	12.15%
LHS, k=2		9.02%	19.79%	-1.76%	10.98%	21.52%	0.43%
LHS, k=3		8.30%	18.62%	-2.01%	12.21%	22.09%	2.33%
CV		31.97%	36.95%	26.99%	32.01%	36.99%	27.03%
PS	s=2	18.85%	25.27%	12.43%	18.85%	25.26%	12.44%
	s=3	22.57%	28.46%	16.67%	22.67%	28.56%	16.79%
	s=4	26.02%	32.23%	19.82%	26.09%	32.29%	19.89%
	s=5	26.21%	31.00%	21.43%	26.36%	31.15%	21.57%

Table 3.39 *Overall Results in Half-length and Standard Deviation by Each Technique for 0.5 Utilization*

- Coefficient of variations at both utilizations indicate that highly loaded systems are more variable as compared to low utilized systems.
- According to the results, in the low utilization level, we observe that CV and PS, which utilize auxiliary variables to reduce the variance, perform better than LHS and AV, which modify the inputs of the simulation to reduce the variance.
- According to the results of the individual applications of VRTs, in the high utilization level, we observe that PS performs worse with respect to the other three methods. Even though CV is the best in half-length improvement, LHS with k=2 achieves the best reduction in variance; however, they are very close to each other.
- Time-in-system is a monotone non-increasing function of interarrival times. Nevertheless, time-in-system is always a monotone increasing function of service times.
- Performance of AV decreases as the utilization or the variability of the system increases. Thus AV is more effective in less utilized (and thus low variable) systems.
- LHS performs better in highly utilized or highly variable systems.
- In general, especially during the applications of AV and LHS, there is no guarantee for reduction in variance. Sometimes, the variance or half-length may rise oppositely (i.e., increase rather than decrease) to the expectations even though the assumptions or requirements of the method are satisfied.
- As the system load or utilization increases, selection of a stratification variate that is highly correlated with the response variable becomes vitally important in PS.
- Increasing the stratification level in PS tends to increase the resulting improvement in VRT. Thus higher stratification levels should be used in order to maximize the return.
- For a selected control variate, CV performs better than PS if there exists a linear association between the control variate and the response variable. This means that knowledge of true mean value should be given priority over the other information extracted from the system. This is illustrated in the Figure 3.7.
- Among the methods inducing correlation among the replications, LHS with k=2 produces the best result at high utilization. However, at the low utilization, AV yields the best result. This is illustrated in Figure 3.8.
- The worst performance is displayed by LHS with k=3 at low utilization and by PS with s=3 when service time is the stratification variate at the high utilization.
- For single systems, the variance reduction by VRTs are shown theoretically for the asymptotic case but this is not guaranteed for simulation runs of finite length due to the slow convergence of variance estimators. In our experiments, we have witnessed some backfires of AV and LHS even in a very long simulation run length of 400,000.

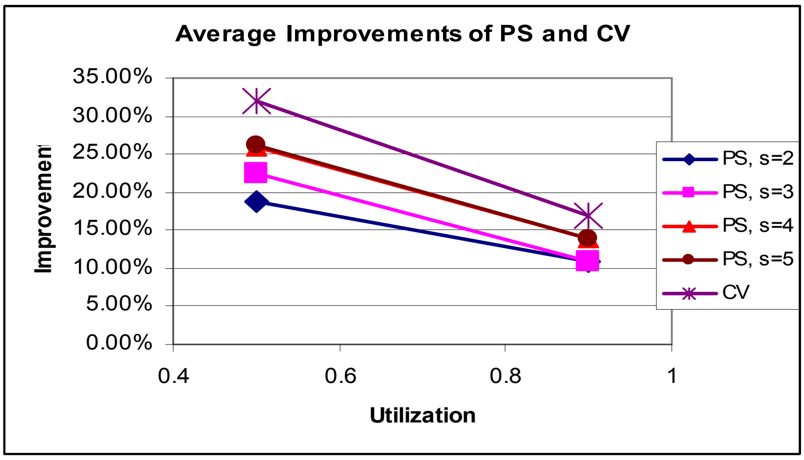

Figure 3.7 *Comparison of CV and PS when Service Time is the Control and the Stratification Variate*

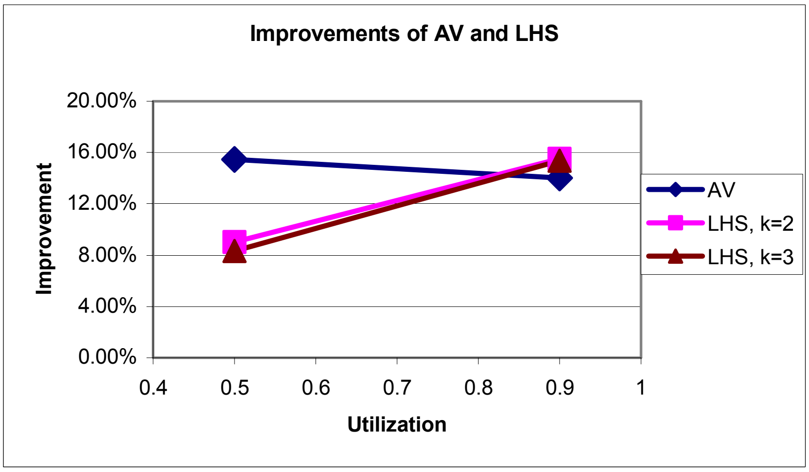

Figure 3.8 *Comparison of AV and LHS*

- Among all the methods CV is the best since it provides the best improvement on the average and its lower bound for the improvement is the highest at both utilizations. Thus CV causes more consistent improvements in variance and the confidence interval.
- PS and CV do not consume much time and effort compared to the simulation run time. Hence, all the alternatives available for CV and PS such as different control variates or higher stratification levels can be tested and the best one is selected for the implementation. In fact this is a significant advantage for PS and CV over AV and LHS.
- Intuitively, since CV and PS modify the output data directly in order to reduce the variance, they are expected to be more efficient with respect to AV and LHS, which modify the input data and thus try to reduce the variance through indirect ways.
- In conclusion, instead of trying to induce a correlation among the replications, the use of auxiliary variables and thus extracting more information from the system is more effective.

3.3. Application of VRTs in Combination

3.3.1. Antithetic Variates + Control Variates (AV+CV)

There are three schemes to combine AV and CV. First way is the direct application of CV to the negatively correlated odd and even numbered runs separately and then combining modified negatively correlated runs in AV. That is, firstly, we generate odd and even replications using AV and then apply CV to each of odd and even series separately. By this way, we correct the time-in-system values in odd and even series. Taking the average of corrected odd and even series, we calculate the standard deviation. Since the degrees of freedom is difficult to calculate, we present the improvement values obtained for the standard deviation for the first scheme.

Second scheme combines the negatively correlated replications before the application of CV. After that, CV is applied to the average of corresponding odd and even replications. Third way employs AV only to produce negatively correlated replications and then assuming all of those replications are generated using independent runs, CV is applied. More explicitly, we don't take an average of odd and even replications. Yang and Liou [40] proved that theoretically third way has the minimum lower bound for the variance. Our results, however, do not comply with the suggestion of Yang and Liou [40]. In Tables 3.40 - 3.45, we present the results when AV is applied to both interarrival and service times and service time is the control variate in CV for 10 different set of experiments.

	AV + CV 1									
	ρ=0.9					**ρ=0.5**				
No	Average	Std. Dev.	a1*	a2*	Improv. (Std. D.)	Average	Std. Dev.	a1*	a2*	Improv. (Std. D.)
1	0.99669	0.002724	89.1843	106.627	**29.19%**	0.200038	0.000077	2.1033	2.7159	**36.50%**
2	0.99825	0.002641	101.3799	109.1536	**31.35%**	0.199878	0.000079	4.2868	4.5508	**34.98%**
3	0.99847	0.002909	127.9281	117.7782	**24.37%**	0.20001	0.000077	4.0383	4.5075	**36.06%**
4	0.99863	0.002991	131.9711	161.2728	**22.24%**	0.199951	0.000085	5.4669	4.3032	**29.31%**
5	1.00377	0.003322	143.0523	93.1272	**13.63%**	0.200055	0.000085	5.5195	4.2849	**29.99%**
6	0.99846	0.003252	98.7792	94.5712	**15.47%**	0.200083	0.000097	4.8432	3.9878	**19.49%**
7	1.00058	0.00242	104.2542	105.2513	**37.08%**	0.200119	0.000068	4.6246	4.8651	**43.56%**
8	0.9993	0.002844	76.0577	131.5008	**26.06%**	0.200107	0.000056	4.0247	4.8879	**53.95%**
9	0.99803	0.002278	69.0813	82.3914	**40.78%**	0.199962	0.000072	3.3648	3.5521	**40.19%**
10	0.99886	0.003478	50.979	127.7469	**9.58%**	0.200015	0.000084	1.4338	4.9947	**30.61%**

Table 3.40 *Results of the First Combination Scheme of AV and CV for both Utilizations*

AV + CV 2 (ρ=0.9)						
No	Average	Std. Dev.	a*	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.995661	0.002660	139.1528	0.005447	29.23%	30.86%
2	0.998348	0.002644	110.8391	0.005416	29.64%	31.26%
3	0.998757	0.002880	82.1585	0.005899	23.37%	25.12%
4	0.999667	0.002863	223.0363	0.005864	23.82%	25.57%
5	1.003438	0.003214	52.5274	0.006583	14.48%	16.44%
6	0.999220	0.003076	-23.3067	0.006300	18.16%	20.03%
7	1.000112	0.002354	146.4598	0.004821	37.36%	38.80%
8	0.998272	0.002902	138.7206	0.005943	22.80%	24.57%
9	0.998081	0.002292	77.2094	0.004695	39.00%	40.40%
10	1.001170	0.003369	173.1243	0.006899	10.37%	12.42%

(Average of a*: 111.9922)

Table 3.41 *Results of the Second Combination Scheme of AV and CV for 0.9 Utilization*

AV + CV 2 (ρ=0.5)						
No	Average	Std. Dev.	a*	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.200002	0.000075	3.8915	0.000154	36.45%	37.91%
2	0.199898	0.000077	5.5709	0.000159	34.35%	35.85%
3	0.200003	0.000077	5.0435	0.000158	34.53%	36.04%
4	0.199945	0.000088	4.7579	0.000180	25.64%	27.35%
5	0.200032	0.000083	5.5929	0.000169	29.95%	31.55%
6	0.200083	0.000096	2.7158	0.000197	18.63%	20.49%
7	0.200100	0.000064	6.5309	0.000131	45.94%	47.18%
8	0.200089	0.000056	5.2370	0.000115	52.19%	53.29%
9	0.199979	0.000072	4.1203	0.000148	38.89%	40.29%
10	0.200123	0.000081	7.1735	0.000165	31.58%	33.15%

(Average of a*: 5.0634)

Table 3.42 *Results of the Second Combination Scheme of AV and CV for 0.5 Utilization*

AV + CV 3 (ρ=0.9)						
No	Average	Std. Dev.	a*	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.996111	0.003104	93.6681	0.006213	19.28%	19.32%
2	0.998253	0.002884	105.3633	0.005775	24.98%	25.02%
3	0.998517	0.002710	123.1879	0.005425	29.53%	29.56%
4	0.998762	0.002895	149.2975	0.005795	24.71%	24.75%
5	1.002953	0.003977	131.5055	0.007962	-3.44%	-3.39%
6	0.998368	0.003367	101.4173	0.006741	12.43%	12.47%
7	1.000579	0.002982	104.7663	0.005970	22.44%	22.48%
8	0.998479	0.003145	94.6222	0.006296	18.21%	18.25%
9	0.998056	0.003123	76.2960	0.006253	18.77%	18.81%
10	0.999937	0.003711	105.6257	0.007429	3.49%	3.54%

Table 3.43 *Results of the Third Combination Scheme of AV and CV for 0.9 Utilization*

AV + CV 3 (ρ=0.5)						
No	Average	Std. Dev.	a*	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.200013	0.000083	2.7588	0.000167	30.87%	30.90%
2	0.199878	0.000089	4.4188	0.000177	26.56%	26.59%
3	0.200007	0.000077	4.3131	0.000154	36.36%	36.40%
4	0.199946	0.000084	4.7971	0.000169	30.07%	30.11%
5	0.200035	0.000098	5.1657	0.000197	18.48%	18.52%
6	0.200072	0.000098	4.3291	0.000197	18.46%	18.50%
7	0.200120	0.000090	4.7250	0.000180	25.42%	25.46%
8	0.200094	0.000071	4.2970	0.000142	41.29%	41.32%
9	0.199962	0.000096	3.4607	0.000193	20.09%	20.13%
10	0.200060	0.000098	3.7147	0.000195	19.16%	19.20%

Table 3.44 *Results of the Third Combination Scheme of AV and CV for 0.5 Utilization*

	AV + CV (ρ=0.9)				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
Scheme 2	**24.82%**	2.92%	31.43%	18.22%	6.60%
Scheme 3	**17.04%**	3.23%	24.35%	9.73%	7.30%
	AV + CV (ρ=0.5)				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
Scheme 2	**34.81%**	3.03%	41.66%	27.97%	6.85%
Scheme 3	**26.68%**	2.52%	32.38%	20.98%	5.70%

Table 3.45 *Confidence Intervals for Improvements by Second and Third Combination Schemes*

These results indicate the superiority of the second scheme over the third scheme in terms of the half-length improvement. This also holds for the standard deviation. The second scheme yields a greater improvement at both utilizations (more than 7%) than the third scheme.

The integration of the AV and CV with any scheme produces better results than the individual applications of the two methods, except for the third scheme at the low utilization. In this case, inducing negative correlation among the replications degrades the improvement obtained by CV alone. Stand-alone application of AV provides the improvements of 14.03% and 15.45% for 0.9 and 0.5 utilizations, respectively, while CV produces 16.95% and 31.97% improvements at 0.9 and 0.5. Thus the best combination scheme increases the improvements of CV alone from 16.95% to 24.93% at the high utilization and from 31.97% to 34.91% at the low utilization. Also, the best scheme increases the improvements of the stand-alone application of AV from 14.03% to 24.93% at high utilization and 15.45% to 34.91% at the low utilization. In conclusion, combining two methods appropriately increases the resulting improvement.

Finally we note that even if we have positive improvements with the first and second schemes, one negative observation is observed in the third scheme at the high utilization. This means inferior results (i.e., backfires) can also result with the good methods but this probability is very small in the first two schemes. As a result, we propose to use the second combination scheme as the best policy for AV and CV combination.

3.3.2. Antithetic Variates + Poststratified Sampling (AV+PS)

In this case, we first take negatively correlated runs by applying AV and then taking the average of each pair, we apply PS to those averages assuming that they are generated using independent replications. However, while constructing the strata, we encounter a difficulty in calculating the standard deviation of the average service time. Since the odd and even replications are not independent, when we take the average of these replications, we need the theoretical covariance between them in order to calculate the standard deviation. Nevertheless, we skip this combination in this and the other two systems.

3.3.3. Latin Hypercube Sampling + Control Variates (LHS+CV)

Since we have obtained the best results with the application of LHS to both interarrival and service times, we apply LHS to both variables and stratify input variables into k=2,3 stratification levels. Also, we take the service time as the control variate. In this combination scheme, CV is directly applied to the macro replications obtained using LHS (i.e. after taking the average of k micro replications), as if they are generated from independent replications. The results for both utilizations are presented in Tables 3.46 - 3.49.

CV + LHS with k=2										
	ρ=0.9					**ρ=0.5**				
No	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.993288	0.002703	0.005535	28.09%	**29.74%**	0.199854	0.000078	0.000161	33.52%	**35.05%**
2	0.996717	0.003005	0.006154	20.06%	**21.89%**	0.199926	0.000106	0.000218	9.97%	**12.03%**
3	0.997504	0.002829	0.005794	24.73%	**26.45%**	0.199918	0.000086	0.000175	27.52%	**29.18%**
4	0.998635	0.002776	0.005685	26.15%	**27.84%**	0.199887	0.000092	0.000188	22.34%	**24.12%**
5	0.997731	0.002355	0.004822	37.35%	**38.79%**	0.199795	0.000065	0.000133	45.09%	**46.35%**
6	0.996122	0.002592	0.005308	31.04%	**32.62%**	0.199886	0.000065	0.000133	44.89%	**46.15%**
7	0.994812	0.002997	0.006138	20.26%	**22.09%**	0.199982	0.000077	0.000159	34.35%	**35.85%**
8	1.000933	0.003396	0.006956	9.63%	**11.71%**	0.200044	0.000085	0.000174	27.88%	**29.54%**
9	0.998379	0.003667	0.007510	2.43%	**4.67%**	0.199950	0.000095	0.000195	19.23%	**21.09%**
10	0.998074	0.002991	0.006125	20.43%	**22.26%**	0.199975	0.000090	0.000185	23.43%	**25.19%**

Table 3.46 *Results of CV and LHS with k=2 for both Utilization Levels*

CV + LHS with k=3										
	ρ=0.9					ρ=0.5				
No	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.996298	0.002420	0.005081	33.99%	**37.10%**	0.199873	0.000085	0.000179	25.88%	**29.37%**
2	1.004196	0.003576	0.007509	2.45%	**7.05%**	0.200044	0.000093	0.000195	19.35%	**23.15%**
3	0.998065	0.002995	0.006289	18.30%	**22.15%**	0.200054	0.000060	0.000127	47.45%	**49.93%**
4	1.000531	0.002978	0.006253	18.76%	**22.59%**	0.199956	0.000073	0.000154	36.29%	**39.30%**
5	0.998528	0.003577	0.007512	2.40%	**7.01%**	0.200038	0.000077	0.000163	32.66%	**35.83%**
6	1.003558	0.002146	0.004506	41.46%	**44.22%**	0.200049	0.000084	0.000177	26.90%	**30.34%**
7	1.004549	0.002141	0.004496	41.59%	**44.34%**	0.200002	0.000092	0.000193	20.26%	**24.02%**
8	1.000471	0.002861	0.006007	21.96%	**25.64%**	0.199843	0.000077	0.000162	33.10%	**36.26%**
9	0.999234	0.003464	0.007275	5.49%	**9.95%**	0.199926	0.000087	0.000183	24.06%	**27.64%**
10	0.997310	0.002639	0.005543	27.99%	**31.39%**	0.199923	0.000084	0.000175	27.40%	**30.82%**

Table 3.47 *Results of CV and LHS with k=3 for both Utilization Levels*

	CV + LHS with k=2				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
ρ=0.9	22.02%	3.20%	29.26%	14.77%	7.24%
ρ=0.5	28.82%	3.49%	36.71%	20.93%	7.89%

Table 3.48 *Confidence Intervals for Improvements by CV and LHS with k=2*

	CV + LHS with k=3				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
ρ=0.9	21.44%	4.72%	32.10%	10.78%	10.66%
ρ=0.5	29.33%	2.65%	35.33%	23.34%	6.00%

Table 3.49 *Confidence Intervals for Improvements by CV and LHS with k=3*

The results indicate that the use of the service time as the control variate and inducing negative correlation among the replications by LHS performs well at the high utilization. A careful examination of the results indicates that the differences between the average improvements are significant at both stratifications. Furthermore, lower limits for the average improvements are very far from zero so this combination produces a considerable reduction in the variance. Also, at the low utilization, this combination results in worse improvements than the stand-alone application of CV.

Increasing the stratification level does not affect the improvement so much. However, it provides more consistent improvements at low utilization while the opposite is true in the other case. Comparing the average improvements of the combined policies with the single application of the LHS indicates the superiority of the combination. In Table 3.50, the improvements in the half-length and the standard deviation by stand-alone application of LHS are given.

	LHS with k=2				LHS with k=3			
	ρ=0.9		ρ=0.5		ρ=0.9		ρ=0.5	
No	Improv. (Half L.)	Improv. (Std. D.)	Improv. (Half L.)	Improv. (Std. D.)	Improv. (Half L.)	Improv. (Std. D.)	Improv. (Half L.)	Improv. (Std. D.)
1	16.26%	18.06%	-10.38%	-8.01%	29.51%	32.51%	19.04%	22.49%
2	12.92%	14.79%	-9.32%	-6.96%	-6.75%	-2.20%	-5.13%	-0.66%
3	19.66%	21.39%	19.26%	21.00%	10.47%	14.28%	24.63%	27.84%
4	9.03%	10.99%	-4.82%	-2.56%	16.59%	20.14%	25.10%	28.29%
5	34.79%	36.20%	33.44%	34.87%	-3.68%	0.74%	-0.36%	3.92%
6	27.69%	29.25%	25.35%	26.95%	39.99%	42.55%	-21.92%	-16.73%
7	16.28%	18.09%	16.11%	17.91%	35.43%	38.18%	10.18%	14.01%
8	4.00%	6.07%	15.26%	17.09%	7.50%	11.44%	6.98%	10.94%
9	-2.31%	-0.11%	3.97%	6.04%	2.42%	6.57%	10.44%	14.25%
10	17.00%	18.79%	1.33%	3.45%	22.06%	25.38%	14.10%	17.75%

Table 3.50 *Individual Improvements in Half-length and Standard Deviation*

Hence, extracting more information via the application of CV after inducing negative correlation increases the reduction in the standard deviation. On the other hand, applying CV sometimes decreases the improvement in the half-length achieved by stand-alone application of LHS at the high utilization. However, this degradation is very small and close to zero in all cases. Comparing the results of the combined policies with the CV individually indicate that on the average, applying CV to independent data results in more improvement at low the utilization. That is, applying CV to negatively correlated replications degrades the precision at the low utilization. The reverse is also true at the high utilization. Hence, applying CV with LHS may not be so beneficial.

3.3.4. Latin Hypercube Sampling + Poststratified Sampling (LHS+PS)

Due to the same reasons stated in Section 3.3.2. Antithetic Variates + Poststratified Sampling, we skip this application as well, in this and the other two systems.

3.3.5. Overview of the Results

We summarize the overall results obtained through the applications of VRTs in Table 3.51 – 3.54. These are itemized as follows:

		0.9					
		Half-length			Standard Deviation		
Method		Improv. (Half L.)	Upper Limit	Lower Limit	Improv. (Std. D.)	Upper Limit	Lower Limit
AV		14.03%	22.53%	5.53%	15.88%	24.19%	7.56%
LHS, k=2		15.53%	23.21%	7.85%	17.35%	24.87%	9.84%
LHS, k=3		15.35%	26.93%	3.78%	17.29%	30.17%	4.41%
CV		16.95%	21.88%	12.01%	16.99%	21.92%	12.05%
PS	s=2	10.86%	16.84%	4.87%	10.95%	16.97%	4.93%
	s=3	10.80%	17.30%	4.30%	10.93%	17.44%	4.41%
	s=4	14.00%	19.43%	8.58%	14.20%	19.63%	8.77%
	s=5	13.83%	19.95%	7.71%	14.03%	20.13%	7.94%

***Table 3.51** Overall Results in Half-length and Standard Deviation by Each Technique for 0.9 Utilization*

		0.5					
		Half-length			Standard Deviation		
Method		Improv. (Half L.)	Upper Limit	Lower Limit	Improv. (Std. D.)	Upper Limit	Lower Limit
AV		15.45%	20.68%	10.22%	17.26%	22.38%	12.15%
LHS, k=2		9.02%	19.79%	-1.76%	10.98%	21.52%	0.43%
LHS, k=3		8.30%	18.62%	-2.01%	12.21%	22.09%	2.33%
CV		31.97%	36.95%	26.99%	32.01%	36.99%	27.03%
PS	s=2	18.85%	25.27%	12.43%	18.85%	25.26%	12.44%
	s=3	22.57%	28.46%	16.67%	22.67%	28.56%	16.79%
	s=4	26.02%	32.23%	19.82%	26.09%	32.29%	19.89%
	s=5	26.21%	31.00%	21.43%	26.36%	31.15%	21.57%

***Table 3.52** Overall Results in Half-length and Standard Deviation by Each Technique for 0.5 Utilization*

		0.9					
		Half-length			Standard Deviation		
Method		Improv. (Half L.)	Upper Limit	Lower Limit	Improv. (Std. D.)	Upper Limit	Lower Limit
AV + CV		24.82%	31.43%	18.22%	26.55%	33.00%	20.09%
LHS + CV	k=2	22.02%	29.26%	14.77%	23.81%	30.88%	16.73%
	k=3	21.44%	32.10%	10.78%	25.14%	35.30%	14.99%

***Table 3.53** Overall Improvements of Combinations in Half-length and Standard Deviation for 0.9 Utilization*

		0.5					
		Half-length			Standard Deviation		
Method		Improv. (Half L.)	Upper Limit	Lower Limit	Improv. (Std. D.)	Upper Limit	Lower Limit
AV + CV		34.81%	41.66%	27.97%	36.31%	43.00%	29.62%
LHS + CV	k=2	28.82%	36.71%	20.93%	30.46%	38.16%	22.75%
	k=3	29.33%	35.33%	23.34%	32.67%	38.38%	26.95%

Table 3.54 *Overall Improvements of Combinations in Half-length and Standard Deviation for 0.5 Utilization*

- In general, combined use of VRTs performs better than the stand-alone applications.
- Using VRTs in combination may sometimes produce worse results than the stand-alone applications of the VRTs. Therefore, an improvement in variance reduction cannot be guaranteed by the combined methods with respect to the stand-alone applications. As an exception, we can give the CV and LHS together at the low utilization.
- The combination of CV with AV provides more improvement than LHS+CV at both utilizations. As in the single application of CV, performance of the combinations also shows a decreasing pattern as the utilization increases. This is illustrated in Figure 3.9.

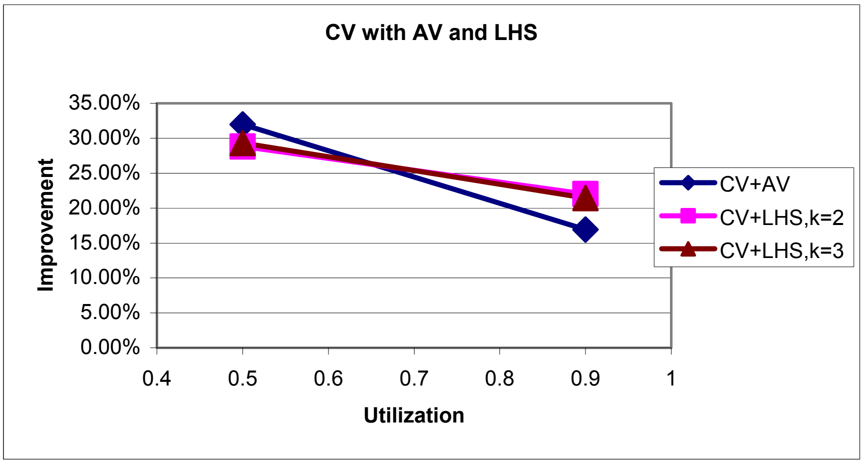

Figure 3.9 *Comparison of CV+AV and CV+LHS*

- The best improvements are produced by the combination AV and CV. This combination contributes much to the improvement achieved by stand-alone applications.
- It seems that increasing the level of stratification improves the obtained reduction even though this is not significant due to the high level of noise.
- Using secondary variables via CV in order to increase the improvement with AV or LHS produces better results than the stand-alone application of AV or LHS.
- Thus the methods using auxiliary variables produce in general better results than the methods modifying the input data (AV and LHS). Yet, it is even better to use these input and output methods simultaneously.

Chapter 4

Analysis of VRTs in the Output Analysis of a Serial Line Production System

4.1. Serial Line Production System of Five Stations

In this chapter, we examine the effects of the four VRTs in a serial line production system of five stations with limited buffers between the stations. Thus the "blockage" and the "starvation" of the stations are allowed. The first station is never starved since there exists an infinite supply of inputs and the last station is never blocked since it can remove all operated parts to the next area after the production line. This serial line production system is illustrated in Figure 4.1, where the squares denote the stations with the circles corresponding to the buffer areas between those stations:

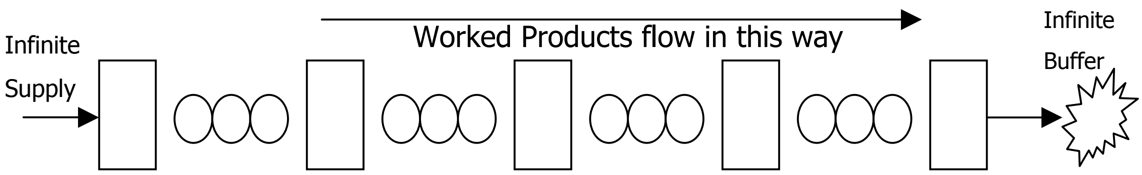

Figure 4.1 *Illustration of the Serial Line Production System of Five Stations*

In this figure, "a", "b", and "c" correspond to the places in the buffer. This system operates as follows: When a station completes its process with a product, it checks whether the buffer next to itself has a free space to place the worked part. It there exists, then the product is transferred to the free space in the buffer and the station starts to work on the next product if there exists. Otherwise, the finished product is waited on the station and no additional product is taken for operation until there occurs a free space in the next buffer. In fact, this case is called the "blockage" of the station. In the "starvation" case, the station is ready to operate but there does not exist any product for operation.

In our experiments, we assume that all stations have the same processing time having a lognormal distribution with a mean of 1 and the standard deviation of 0.3. This is actually one of the mostly considered cases in the literature. As stated before, we use "throughput" of the system as the performance measure. The following individual and combined VRTs are applied on the throughput of the production system:

Serial Line Production System	
AV	PS (s=2, 3, 4, 5)
LHS (k=2, 3)	AV+CV
CV	LHS (k=2, 3) +CV

Table 4.1 *List of Single and Integrated VRTs that will be Applied to Serial Line Production System*

In order to determine the warm up period, a pilot simulation runs are taken and the graph of throughput versus the number of entities is plotted as follows:

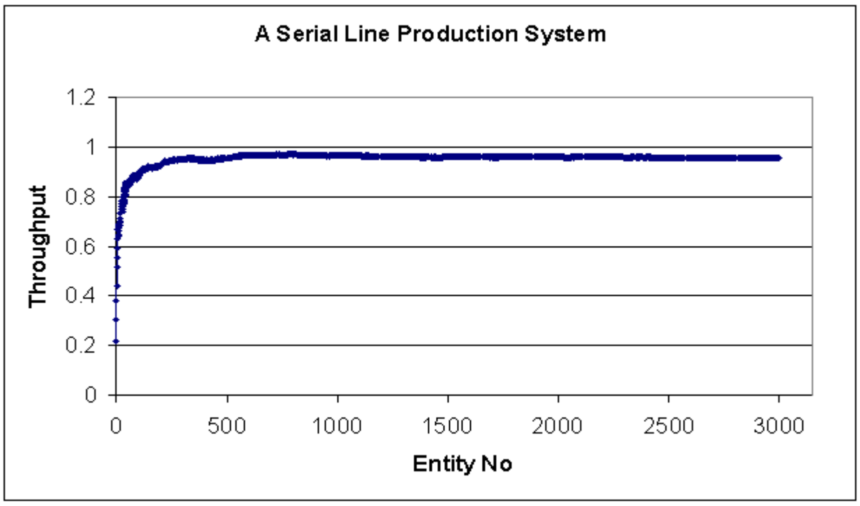

Figure 4.2 *Warm up Period for the Serial Line Production System*

According to this graph, the warm up period is estimated to be around the first 500 product completions. However, in order to be on the safe side, we discard the first 800 and calculate the corresponding statistics for the next 40,000 products. Again, we take independent 60 replications in each case with the same reasoning stated in the M/M/1 system. We construct confidence intervals on the improvements in half-length and standard deviation via 10 different sets of experiments. The results of these 10 experiments are used during the performance evaluation.

4.2. Application of VRTs Individually

4.2.1. Independent Case

Since the only input variable to this system is the service times, in each of 10 experiments, we use different service times. We present the results of 10 experiments together with the stream numbers and the initial seeds in Table 4.2. Using the results of ten experiments, benchmark values are calculated for half-length and standard deviation and are presented in Table 4.3.

Independent Case					
No	Seed	Mean	Std. Dev.	CV	Half-length
1	1	0.955612	0.000126	0.000132	0.000252
2	2	0.955555	0.000123	0.000129	0.000246
3	3	0.955723	0.000118	0.000124	0.000236
4	4	0.955947	0.000111	0.000117	0.000223
5	5	0.956006	0.000114	0.000120	0.000229
6	6	0.955701	0.000120	0.000125	0.000240
7	7	0.955929	0.000116	0.000122	0.000233
8	8	0.956004	0.000125	0.000130	0.000250
9	1, 23543	0.955724	0.000107	0.000112	0.000215
10	2, 11287	0.955769	0.000132	0.000138	0.000263

Table 4.2 *Results of the Ten Independent Replications with the Stream and Initial Seed Numbers*

	Estimated Half-length and Standard Deviation				
	Average	Std.Dev.	Upper Limit	Lower Limit	Half-length
Half-length	**0.000239**	0.000005	0.000242	0.000235	0.000003
Std. Dev.	**0.000119**	0.000002	0.000121	0.000118	0.000002

Table 4.3 *Benchmark Half-length and Standard Deviation Values*

4.2.2. Antithetic Variates (AV)

Since the inputs to the serial line production system are the service times of stations, we apply AV to each service time simultaneously based on the experience with the M/M/1 system. We repeat this procedure ten times using different streams and initial seeds. 30 pairs of replications have been taken in the construction of AV. The results are presented Tables 4.4 and 4.5.

In general, AV when applied to each service time simultaneously in a serial production system of five stations provides an improvement of 7.61% in the half-length on the average. However, since the confidence interval of the half-length include the zero, we conclude that there is no significant improvement in precision by the application of AV. As seen in the individual improvement values, sometimes even the worse results are observed. Nevertheless, in accordance with our experimental results, the success rate of AV can be approximated as 80%, i.e. in eight of ten cases a positive improvement is observed. Oppositely, since the confidence interval of the standard deviation does not include zero, AV can guarantee success in terms of the standard deviation even though its lower limit is very close to zero.

AV to All Service Times								
No	Std. Dev.	Covariance	Correlation	Var(ODD)	Var(EVEN)	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.000115	0.00000002	0.032471	0.000001	0.000001	0.000234	1.88%	3.99%
2	0.000076	-0.00000037	-0.523124	0.000001	0.000001	0.000154	35.31%	36.70%
3	0.000107	-0.00000040	-0.375882	0.000001	0.000001	0.000219	8.22%	10.20%
4	0.000136	0.00000001	0.007096	0.000001	0.000001	0.000278	-16.50%	-13.99%
5	0.000103	-0.00000031	-0.333905	0.000001	0.000001	0.000211	11.59%	13.49%
6	0.000103	-0.00000050	-0.475314	0.000001	0.000002	0.000210	12.09%	13.98%
7	0.000101	-0.00000029	-0.318693	0.000001	0.000001	0.000207	13.20%	15.06%
8	0.000122	-0.00000012	-0.124210	0.000001	0.000001	0.000250	-4.89%	-2.63%
9	0.000103	-0.00000030	-0.325981	0.000001	0.000001	0.000211	11.48%	13.39%
10	0.000112	-0.00000028	-0.267868	0.000001	0.000001	0.000230	3.69%	5.77%

Table 4.4 *Results of AV applied to All Service Times*

	Confidence Intervals				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
Half-length	**7.61%**	4.26%	17.24%	-2.02%	9.63%
Std. Dev.	**9.60%**	4.17%	19.02%	0.17%	9.43%

Table 4.5 *Confidence Intervals for Improvements when AV applied to All Service Times*

As stated before, these are the results of the comparison of AV with the benchmark half-length and standard deviation values. However, we also compare the results of the AV with the corresponding independent runs. This comparison obeys the logic of the VRTs. That is after taking 30 independent runs, we decide whether to generate 30 additional runs independently of the previous 30 runs or to construct the complementary runs of the previous 30 independent runs. In Tables 4.6 and 4.7, we present the results of this comparison in terms of estimated half-lengths and standard deviation.

An interesting observation is that in first and fourth cases, positive correlations are observed between the so-called ‘negatively correlated replications’. However, even with this positive correlation, improvements in the half-length and the standard deviation are obtained. Oppositely, in the eighth case, even though a negative correlation exists between the odd and even numbered replications, negative improvements are obtained both in terms of the half-length and the standard deviation. These unexpected results emerge as a result of working with finite sample as in M/M/1. They indicate that the variances of odd and even numbered replications have not converged to the same value. Hence, longer runs are required so that the variances reach the steady state.

No	Half-length			Standard Deviation		
	Independent	AV	Improv.	Independent	AV	Improv.
1	0.000252	0.000234	7.10%	0.000126	0.000115	9.10%
2	0.000246	0.000154	37.34%	0.000123	0.000076	38.68%
3	0.000236	0.000219	7.34%	0.000118	0.000107	9.33%
4	0.000223	0.000278	-24.76%	0.000111	0.000136	-22.07%
5	0.000229	0.000211	7.81%	0.000114	0.000103	9.80%
6	0.000240	0.000210	12.47%	0.000120	0.000103	14.35%
7	0.000233	0.000207	10.97%	0.000116	0.000101	12.89%
8	0.000250	0.000250	-0.32%	0.000125	0.000122	1.84%
9	0.000215	0.000211	1.61%	0.000107	0.000103	3.73%
10	0.000263	0.000230	12.73%	0.000132	0.000112	14.61%

Table 4.6 *Individual Improvements in Half-length and Standard Deviation by AV*

	Confidence Intervals				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
Half-length	**7.23%**	4.82%	18.12%	-3.66%	10.89%
Std. Dev.	**9.23%**	4.72%	19.88%	-1.43%	10.66%

Table 4.7 *Confidence Intervals for Improvements in Half-length and Standard Deviation by AV*

In conclusion, AV does not guarantee an improvement in variance reduction and in precision. The vital requirement of AV to reduce the variance is the monotone relationship between the performance measure and the related input random variables. In a serial line system, throughput is mostly determined by the ‘bottleneck’ station, which is the one in the middle when all service times are equal as in our case. An analogy can be established with a pipeline system transferring water between two places and having different intersection area at different points. Therefore, the maximum amount of water flowing in this pipeline per unit time equals to the maximum amount that can flow in the narrowest point of this pipeline per unit time. Based on this analogy, it can be induced that throughput is a monotone non-increasing function of the service times of each station. Plateaus in the throughput vs. service times of each station occur when the current station is no longer the bottleneck and after that point, a plateau belonging to the new bottleneck station ends and a decreasing part emerges. As a result, when all of the stations are taken into consideration, it is seen that throughput is a monotone function of the combination of all service times of stations.

4.2.3. Latin Hypercube Sampling (LHS)

We consider k=2, 3 stratification levels again. Based on the experience with M/M/1, we apply LHS to the service times of all stations simultaneously. The streams and initial seeds in Table 4.8 are used to generate permutations and service times for k=2 and k=3. Results are presented in Tables 4.9 - 4.11 for both stratification levels.

No	1	2	3	4	5	6	7	8	9	10
Permutation	1	3	5	7	1-23251	3-19631	5-31235	7-23157	1-23591	3-11233
Service Times	2	4	6	8	2-13571	4-17913	6-12357	8-11537	2-15423	4-21357

Table 4.8 *Stream and Initial Seed Numbers for LHS*

No	LHS with k=2					LHS with k=3				
	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.955727	0.00011	0.000226	5.44%	7.47%	0.955498	0.000115	0.00024	-0.69%	3.59%
2	0.955592	0.000136	0.000278	-16.38%	-13.88%	0.955677	0.000106	0.000222	7.11%	11.06%
3	0.955631	0.000109	0.000224	6.31%	8.32%	0.955956	0.000135	0.000282	-18.31%	-13.27%
4	0.956097	0.000134	0.000274	-14.63%	-12.16%	0.955812	0.00011	0.000231	3.24%	7.36%
5	0.955973	0.000114	0.000234	2.14%	4.24%	0.955943	0.000121	0.000254	-6.23%	-1.71%
6	0.95592	0.000101	0.000207	13.19%	15.06%	0.955758	0.000105	0.000219	8.42%	12.32%
7	0.95582	0.000104	0.000213	10.89%	12.81%	0.955831	0.000078	0.000163	31.71%	34.61%
8	0.955742	0.000102	0.000209	12.28%	14.17%	0.955866	0.000093	0.000195	18.31%	21.79%
9	0.955718	0.000085	0.000173	27.58%	29.14%	0.955828	0.000098	0.000205	14.23%	17.88%
10	0.955837	0.000094	0.000193	19.05%	20.79%	0.955933	0.000101	0.00021	11.90%	15.65%

Table 4.9 *Results of LHS with k=2 when applied to All Service Times*

	Half-length				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
LHS, k=2	**6.59%**	4.33%	16.38%	-3.21%	9.79%
LHS, k=3	**6.97%**	4.35%	16.81%	-2.87%	9.84%

Table 4.10 *Confidence Intervals for Improvements in Half-length by LHS with k=2 and k=3*

	Standard Deviation				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
LHS, k=2	**8.60%**	4.24%	18.18%	-0.98%	9.58%
LHS, k=3	**10.93%**	4.17%	20.35%	1.51%	9.42%

Table 4.11 *Confidence Intervals for Improvements in Standard Deviation by LHS with k=2 and k=3*

The results indicate that k=2 and k=3 stratification levels do not differ from each other in terms of the half-lengths. They both have nearly not only the same average improvement and half-length but also the same success rate. LHS with k=2 failed in two instances out of ten while LHS with k=3 failed in three instances, one of which is very close to zero.

Conversely, k=2 and k=3 differ in terms of the improvements in standard deviation. LHS with k=3 produced a larger improvement on the average and its confidence interval does not include zero unlikely to k=2. This means that although LHS with k=3 somehow guarantees reduction in the variance LHS with k=2 does not. Moreover, individual improvements indicate that negative improvements can still be found in both cases. The difference of the conclusions for half-length and standard deviation originates from the differences in t value due to the degrees of freedom. 19 degrees of freedom is used at k=3 and 29 at k=2.

The difference between the average improvements in half-length belonging to two stratification levels is too small. Despite this slight difference, a significant difference is observed in terms of standard deviation. Therefore, it seems that as the level of stratification increases, the amount of improvements increases. We summarize the results in Table 4.12.

	AV	LHS, k=2	LHS, k=3
Half-length	7.61%	6.59%	6.97%
Standard Deviation	9.60%	8.60%	10.93%

Table 4.12 *Comparison of AV and LHS in terms of Half-length and Standard Deviation Improvement*

In this table, we note a slight differences between the improvements in the half-lengths while AV seems the best. In terms of the improvements in the standard deviation, LHS with =3 appears to be the best. Once again, the difference between the AV and LHS with k=3 may be negligible. Nevertheless, LHS with k=2 is the worst of three in both cases. Obviously, the improvements in half-lengths are smaller with respect to standard deviations since during the calculation of half-lengths, we loose degrees of freedoms.

4.2.4. Control Variates (CV)

In this case, ‘service times’ of all stations assuming that we know the theoretical or expected means, are chosen as our control variates. Since the same service rate is used in all stations, we used the ‘theoretically known mean’ of service time as 1 minute in all cases. Prior to the simulation we are aware of the non-linear dependence between the throughput and the service time. Nevertheless, assuming that there exists a linear association, we apply CV to this case.

During the implementation of CV when all service times are used as control variates, we use t value with degrees of freedom 54 instead of 58 since the control coefficients are estimated five times for each station. The results are presented in Table 4.13.

	No	Service1	Service 2	Service3	Service4	Service5	ALL
	1	-1.74%	5.68%	-1.13%	-2.79%	-1.26%	27.34%
	2	3.64%	6.96%	3.07%	3.78%	2.33%	41.14%
	3	3.51%	7.58%	13.18%	11.87%	14.98%	33.68%
	4	8.90%	9.21%	9.11%	12.24%	13.18%	30.27%
	5	4.57%	5.43%	13.30%	15.08%	10.36%	36.45%
	6	1.31%	8.28%	17.56%	6.36%	1.04%	32.98%
	7	13.98%	5.87%	13.10%	18.70%	2.88%	36.44%
	8	3.33%	2.72%	-0.16%	1.85%	1.04%	26.67%
	9	10.18%	10.81%	20.74%	17.56%	11.76%	35.15%
	10	-4.40%	-4.18%	0.57%	-6.71%	-7.02%	23.07%
Confidence Intervals for Different Control Variates	Average	**4.33%**	**5.84%**	**8.94%**	**7.79%**	**4.93%**	**32.32%**
	Std. Dev.	1.74%	1.32%	2.49%	2.75%	2.28%	1.73%
	Upper Limit	8.26%	8.82%	14.56%	14.00%	10.08%	36.23%
	Lower Limit	0.39%	2.85%	3.32%	1.58%	-0.22%	28.41%
	Half-length	3.93%	2.98%	5.62%	6.21%	5.15%	3.91%

Table 4.13 *Results of CV and Confidence Intervals for Improvements in Half-length when Different Service Times are used as Control Variates*

The results presented here are in terms of half-lengths. Improvements in standard deviation are larger than these values since there occurs a loss in the half-lengths due to degrees of freedom. Nevertheless, we present the confidence interval for the average improvement in the standard deviation when all service times are used as control variates in Table 4.14.

Confidence Interval in terms of the Improvement in the Standard Deviation for 'ALL' case				
Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
32.46%	1.72%	36.36%	28.57%	3.90%

Table 4.14 *Confidence Intervals for Improvements in Standard Deviation when All Service Times are taken as Control Variates*

The results indicate that choosing any service time as the control variate provides an improvement and the amount of this varies according to the station. As the expected bottleneck in the system, middle station provides more improvement compared to other stations. This implies that the throughput is mostly affected by the middle of the serial line production system. A decreasing effect is observed while getting away from the middle. Moreover, selecting the service time of a station through the end of the production system performs better than selecting it in the beginning of the system. This resembles a "bowl" shape that indicates that bottleneck mostly remains in the middle and it is less likely towards either direction.

In addition, as seen in Table 4.13, when CV is applied to the all service times, an improvement level, which is slightly larger than the sum of individual improvement levels, is observed on the average. The sum of five averages is 31.82% while the result of 'ALL' case is 32.32%. Nevertheless, this observation is not valid for the ten individual cases. In some cases, the sum of individual improvements turns out to be smaller than the improvement of 'ALL' case and sometimes, the reverse becomes true. A very interesting observation occurs in the tenth case. Even though individual improvement values are all negative except one and summing to –21.75%, when all service times are used as control variates an improvement level of 23.07% is obtained. The explanation is that when CV is applied to all service times simultaneously, there exists a higher linear association between the service times and the throughput.

Furthermore, the use of all service times as the control variates results in better improvements than the use of individual service times as control variates. Hence, CV should be applied to all possible input variables that are predicted to be pursuing a linear relationship with the performance measure. Also, there exist negative improvement values when the service times are used as control variates individually. However, all of the improvement values in the 'ALL' case are substantially larger than zero. As a result, when all service times are used as control variates, an improvement between 28.41% - 36.23% is expected with 95% probability.

When CV is compared with the previously discussed two methods, AV and LHS, CV achieves more improvement in both half-lengths and standard deviation. Therefore, instead of inducing a negative correlation among the replications, applying CV and modifying the output data would be much better in VRT applications.

4.2.5. Poststratified Sampling (PS) (s=2,3,4,5)

For this method, many alternatives can be considered since each service time can be selected individually as stratification variate in addition to flow time or time spent in the buffers. However, only one of them should be used in the implementation of PS. We select the one with the highest correlation with the throughput, which seems as the service time of the third station (i.e., the middle station). As explained before, the bottleneck station mostly determines throughput of the serial production systems when the service times of the stations are equal. Furthermore, the non-linear association between the service time and the throughput would not constitute any problem since PS can be used successfully in these cases in contrast to CV.

We consider four different stratification levels s=2,3,4,5. Again, we form each stratum according to the equal probability scheme and the optimal allocation scheme proposed by Sethi [33]. Similarly to the previous cases, we perform 10 sets of simulation experiments using different random number streams. The results are presented in Table 4.15 and 4.16.

Improvement Levels obtained with Optimal Allocation Scheme									
	Half-length Improvement					Standard Deviation			
No	s=2	s=3	s=4	s=5	No	s=2	s=3	s=4	s=5
1	-2.80%	-4.54%	-1.94%	-2.31%	**1**	-1.55%	-3.16%	-0.74%	-1.14%
2	-0.62%	1.79%	2.35%	2.48%	**2**	0.60%	3.24%	3.66%	4.13%
3	9.98%	11.76%	10.50%	8.95%	**3**	11.08%	12.77%	11.65%	10.53%
4	8.12%	11.59%	8.61%	11.59%	**4**	9.31%	12.73%	10.06%	12.75%
5	11.49%	13.20%	13.93%	9.26%	**5**	12.60%	14.20%	15.07%	10.33%
6	8.54%	11.38%	14.73%	12.19%	**6**	9.69%	12.79%	15.95%	13.65%
7	14.96%	9.17%	13.59%	11.80%	**7**	16.00%	10.32%	14.58%	12.83%
8	-3.40%	-3.05%	-2.60%	2.52%	**8**	-2.11%	-1.70%	-1.33%	4.01%
9	16.08%	16.78%	16.20%	18.06%	**9**	17.10%	17.74%	17.44%	19.18%
10	-2.26%	-4.81%	-3.71%	-4.09%	**10**	-1.01%	-3.52%	-2.46%	-2.88%

Confidence Intervals for Different Stratification Levels									
	Average	**6.01%**	**6.33%**	**7.17%**	**7.05%**	**7.17%**	**7.54%**	**8.39%**	**8.34%**
	Std. Dev.	2.40%	2.58%	2.49%	2.24%	2.37%	2.54%	2.48%	2.23%
	Upper Limit	11.43%	12.15%	12.80%	12.11%	12.52%	13.27%	13.99%	13.37%
	Lower Limit	0.60%	0.50%	1.53%	1.98%	1.82%	1.81%	2.78%	3.31%
	Half Length	5.41%	5.83%	5.63%	5.07%	5.35%	5.73%	5.61%	5.03%

Table 4.15 *Results of PS and Confidence Intervals for Improvements at each Stratification Level obtained with Optimal Allocation Scheme*

Improvement Levels obtained with Equal Probability Scheme									
	Half-length Improvement					Standard Deviation			
No	s=2	s=3	s=4	s=5	No	s=2	s=3	s=4	s=5
1	-2.80%	-2.00%	-2.84%	-2.79%	**1**	-1.55%	-0.76%	-1.59%	-1.61%
2	-0.62%	1.96%	1.44%	-0.72%	**2**	0.60%	3.34%	2.83%	0.83%
3	9.98%	10.63%	11.47%	10.13%	**3**	11.08%	11.72%	12.61%	11.25%
4	8.12%	12.11%	11.03%	8.46%	**4**	9.31%	13.24%	12.48%	9.68%
5	11.49%	12.29%	13.64%	14.42%	**5**	12.60%	13.33%	14.69%	15.43%
6	8.54%	10.84%	11.75%	11.50%	**6**	9.69%	12.17%	12.92%	13.31%
7	14.96%	9.49%	13.77%	13.09%	**7**	16.00%	10.56%	14.98%	14.09%
8	-3.40%	-2.27%	-3.97%	-2.75%	**8**	-2.11%	-0.91%	-2.60%	-1.30%
9	16.08%	18.04%	15.29%	15.41%	**9**	17.10%	19.00%	16.42%	16.67%
10	-2.26%	-1.85%	-3.55%	-3.01%	**10**	-1.01%	-0.61%	-2.33%	-1.60%

Confidence Intervals for Different Stratification Levels									
	Average	**6.01%**	**6.92%**	**6.80%**	**6.37%**	**7.17%**	**8.11%**	**8.04%**	**7.67%**
	Std. Dev.	2.40%	2.31%	2.53%	2.45%	2.37%	2.28%	2.51%	2.43%
	Upper Limit	11.43%	12.15%	12.52%	11.92%	12.52%	13.25%	13.71%	13.16%
	Lower Limit	0.60%	1.70%	1.08%	0.83%	1.82%	2.96%	2.37%	2.19%
	Half Length	5.41%	5.22%	5.72%	5.54%	5.35%	5.15%	5.67%	5.49%

Table 4.16 *Results of PS and Confidence Intervals for Improvements at each Stratification Level obtained with Equal Probability Scheme*

According to the results of both schemes, as the level of stratification increases, the resulting improvement may not increase. Moreover, even though negative improvements are observed, all of the confidence intervals exclude zero. Thus PS is expected to provide some improvement in the precision and the variance. Optimal allocation scheme performs better when the stratification variate shows the behaviour of a normal distribution. In this case, service time averages of the third station already display this pattern due to the Central Limit Theorem.

In the M/M/1 case, CV has outperformed PS when the same random variable is used as the control and the stratification variate. The same observation is valid in this case when the service time of the third station is used as both the control and the stratification variate. This implies that the knowledge of the theoretical mean of the service time dominates the non-linear dependence between the service time and the throughput. A very obvious advantage of CV over PS is that PS can use only one random variable as the stratification variate while CV can use as many as required. In this case, the feature of CV helped itself to result in a very large improvement both in the half length and the standard deviation and thus to overcome the PS.

In this case, variance reduction does not indicate a clear improvement pattern with the increasing level of stratification. Furthermore, since a non-linear relationship exists between the service time and the throughput, PS should outperform CV in this case as opposed to the M/M/1 case. Nevertheless, the converse is observed since CV can use more than one variable as the control variates while PS cannot.

4.2.6. Overview of the Results

We summarize the overall results obtained through the applications of each VRT in Table 4.17:

		Serial Line Production System					
		Half Length			Standard Deviation		
Method		Improv. (Half L.)	Upper Limit	Lower Limit	Improv. (Std. D.)	Upper Limit	Lower Limit
AV-ALL		7.61%	17.24%	-2.02%	9.60%	19.02%	0.17%
LHS, k=2		6.59%	16.38%	-3.21%	8.60%	18.18%	-0.98%
LHS, k=3		6.97%	16.81%	-2.87%	10.93%	20.35%	1.51%
CV-Service Time 1		4.38%	8.31%	0.45%	4.38%	8.31%	0.45%
CV-Service Time 2		5.89%	8.87%	2.91%	5.89%	8.87%	2.91%
CV-Service Time 3		8.99%	14.60%	3.37%	8.99%	14.60%	3.37%
CV-Service Time 4		7.84%	14.05%	1.64%	7.84%	14.05%	1.64%
CV-Service Time 5		4.98%	10.13%	-0.16%	4.98%	10.13%	-0.16%
CV-ALL		32.32%	36.23%	28.41%	32.46%	36.36%	28.57%
PS- Service Time 3	s=2	6.01%	11.43%	0.60%	7.17%	12.52%	1.82%
	s=3	6.33%	12.15%	0.50%	7.54%	13.27%	1.81%
	s=4	7.17%	12.80%	1.53%	8.39%	13.99%	2.78%
	s=5	7.05%	12.11%	1.98%	8.34%	13.37%	3.31%

Table 4.17 *Overall Improvements in Half-length and Standard Deviation by Each Technique alone*

- Using auxiliary variables and thus extracting more information from the system is again more effective in terms of the resulted variance reduction and precision. That is, the use of auxiliary variables provides better improvements compared to VRTs which manipulates the input random numbers.

- CV is the best since it allows the usage of more than one variable as the control variate.
- Compared to the M/M/1 system, the positive effect of VRTs (AV and LHS) which control the input data diminishes in the serial line case because of the fact that this effect is reduced due to the filtering by the system. This has important practical implications; it suggests that one should concentrate on methods using auxiliary variables (CV and PS).
- The improvements achieved by the application of CV to service times individually resemble a "bowl" shape phenomena. The reason is that bottleneck mostly occurs in the middle thus the service time of the middle station tells more about the throughput of the line. Also, this affect gets smaller towards the beginning and end of the line.
- The improvement achieved by the application of CV to all service times is nearly equal to the sum of the individual achievements on the average. Therefore, CV should be applied to all available input random variables which are anticipated to have or to imply a linear relationship with the performance measure and have a known mean.
- In particular, CV performs better than PS for a selected control variate. In general if there exists a linear association between the control variate and the response variable, this result is expected. In the serial line production system case, CV again outperforms PS even though the association between the service time of the third station and the throughput is non-linear. In fact, PS does not perform well in this case.
- In general, methods inducing correlation (AV and LHS) do not require any additional knowledge about the system. Therefore, if it is difficult to find the relationship between the output variables to extract more information about the system, we can easily apply methods modifying the input random variables.
- Among the methods inducing correlation, LHS and AV, do not differ in terms of the average improvements in the half-length and the corresponding lower and upper limits. Nevertheless, LHS with k=3 performs better in the reduction of variance or standard deviation compared to AV and LHS with k=2.
- Consistently with the M/M/1 system, AV is applied to all service times. However, the confidence interval for the average improvement in half-length includes zero. This means that AV may not improve half-length. Similarly, LHS does not provide any significant improvement in half-length.
- Confidence intervals for the average improvements in half-length with LHS include zero at both levels. This means that no improvement can be obtained with LHS. This deduction is also valid at k=2 for the improvement in the variance.

4.3. Application of VRTs in Combination

4.3.1. Antithetic Variates + Control Variates (AV+CV)

While combining AV and CV, we try the three different schemes explained in the M/M/1 case. AV is applied to each service time simultaneously and CV is applied by choosing any service time or all service times as control variates. In the first scheme we present the improvements in the standard deviation when CV is applied to all service times as in the M/M/1 case. In the other two cases the results are presented in terms of half-length and standard deviation. In addition, degrees of freedom is taken as 24 and 54 in the second and third combination schemes. The results are presented in Tables 4.18 - 4.22.

	Individual Improvements for the First Combination Scheme									
No	1	2	3	4	5	6	7	8	9	10
Std. Dev.	14.07%	49.32%	20.45%	8.34%	29.21%	27.63%	34.33%	7.46%	26.11%	16.79%

Table 4.18 *Individual Improvements in Standard Deviation for the First Scheme*

	Confidence Interval for the First Scheme				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
Std. Dev.	**23.37%**	4.04%	32.51%	14.24%	9.14%

Table 4.19 *Confidence Interval for the Improvements in Standard Deviation by the First Scheme*

	Second Combination Scheme					
No	Service1	Service 2	Service3	Service4	Service5	ALL
1	8.12%	3.92%	8.43%	4.32%	3.77%	12.67%
2	37.66%	46.18%	39.33%	35.41%	35.41%	51.39%
3	25.50%	10.01%	8.30%	11.60%	9.61%	30.39%
4	-14.54%	-13.47%	-14.35%	-12.62%	-1.58%	6.65%
5	11.98%	13.98%	14.58%	13.82%	14.19%	21.64%
6	13.45%	13.18%	12.19%	16.00%	26.49%	30.36%
7	13.11%	15.63%	17.32%	18.20%	19.98%	33.62%
8	-3.62%	-3.67%	-1.16%	-3.95%	-4.78%	2.20%
9	11.53%	17.90%	19.96%	13.96%	17.39%	30.44%
10	3.60%	6.60%	11.22%	4.45%	3.63%	16.41%

Confidence Intervals for Different Control Variates	Average	**10.68%**	**11.02%**	**11.58%**	**10.12%**	**12.41%**	**23.58%**
	Std. Dev.	4.55%	4.95%	4.40%	4.15%	4.03%	4.63%
	Upper Limit	20.95%	22.21%	21.52%	19.50%	21.52%	34.05%
	Lower Limit	0.40%	-0.16%	1.65%	0.74%	3.30%	13.11%
	Half-length	10.28%	11.19%	9.93%	9.38%	9.11%	10.47%

Table 4.20 *Results of the Second Combination Scheme and Confidence Intervals for Improvements When Different Service Times are taken as Control Variates*

	Third Combination Scheme					
No	Service1	Service 2	Service3	Service4	Service5	ALL
1	7.16%	10.02%	9.35%	9.45%	6.83%	26.33%
2	6.44%	13.25%	21.07%	14.43%	9.17%	44.45%
3	-11.13%	-5.57%	7.38%	4.89%	6.35%	13.22%
4	-6.68%	-7.77%	-4.04%	1.00%	-9.34%	17.42%
5	-3.31%	-3.95%	2.15%	4.97%	4.09%	24.44%
6	-13.93%	-4.88%	6.44%	-8.15%	-8.30%	27.01%
7	2.87%	2.93%	15.85%	10.65%	-1.17%	34.08%
8	-3.77%	0.77%	-2.34%	-7.24%	-4.99%	16.36%
9	-2.13%	-1.20%	19.04%	7.06%	-3.98%	26.29%
10	-7.88%	-6.60%	3.97%	-3.74%	-3.46%	20.08%

Confidence Intervals for Different Control Variates	Average	**-3.24%**	**-0.30%**	**7.89%**	**3.33%**	**-0.48%**	**24.97%**
	Std. Dev.	2.24%	2.26%	2.71%	2.43%	2.10%	2.91%
	Upper Limit	1.82%	4.81%	14.01%	8.83%	4.26%	31.55%
	Lower Limit	-8.29%	-5.41%	1.76%	-2.16%	-5.22%	18.38%
	Half-length	5.06%	5.11%	6.13%	5.50%	4.74%	6.58%

Table 4.21 *Results of the Third Combination Scheme and Confidence Intervals for Improvements When Different Service Times are taken as Control Variates*

	Confidence Intervals for Improvements in Standard Deviation											
	Second Combination Scheme						Third Combination Scheme					
	Serv. 1	Serv. 2	Serv. 3	Serv. 4	Serv. 5	ALL	Serv. 1	Serv. 2	Serv. 3	Serv. 4	Serv. 5	ALL
Average	**12.73%**	**13.07%**	**13.61%**	**12.18%**	**14.42%**	**25.91%**	**-3.18%**	**-0.25%**	**7.93%**	**3.38%**	**-0.43%**	**25.15%**
Std. Dev.	4.44%	4.84%	4.29%	4.06%	3.94%	4.49%	2.24%	2.26%	2.71%	2.43%	2.10%	2.91%
Upper Limit	22.77%	24.00%	23.32%	21.35%	23.32%	36.06%	1.87%	4.86%	14.06%	8.87%	4.31%	31.72%
Lower Limit	2.69%	2.14%	3.91%	3.02%	5.52%	15.76%	-8.24%	-5.36%	1.81%	-2.11%	-5.17%	18.59%
Half-length	10.04%	10.93%	9.70%	9.17%	8.90%	10.15%	5.06%	5.11%	6.12%	5.49%	4.74%	6.57%

Table 4.22 *Confidence Intervals for Improvements in Standard Deviation by CV When Different Service Times are taken as Control Variates*

As it is proved by Yang and Liou (1996), the third scheme finds the minimum lower bound on the variance but our findings do not comply with this result. The results indicate that when CV is applied to 'ALL' service times, the third scheme is the best. Therefore, it may be preferred over the other two methods. The basic idea is that more variable observation values tell more about the system. In the third scheme, negatively correlated replications allow the evaluation of the behavior in more extreme cases since it does not smooth out the data by averaging the negatively correlated replications, which destroys those extreme or more variable cases. Hence, more radical changes on the response variable are made. However, average improvements are very close in the second and third schemes. Thus second scheme can still be used securely. Moreover, first scheme is the worst of three in the variance improvement.

Even though in the second scheme all averages are positive and all confidence intervals except one exclude zero, in the third scheme three of the five averages are negative and all confidence intervals except one include zero. Therefore, the third scheme fails when CV is applied to individual service times and the second scheme is more robust than the third scheme. Moreover, the sum of individual values is larger than the result of the combination in the second scheme while the opposite is true in the third scheme.

In all three schemes, combining AV with CV degrades the improvement in half-length or standard deviation obtained when only CV is applied to all service times. CV alone produces an improvement of 32.46% in the standard deviation while the combinations with AV produce 23.37%, 25.91%, and 25.15%. Conversely, when CV is applied to service times individually, we note that even though the previous conclusion holds for the third scheme, combining CV with AV in the second scheme contributes positively to the improvements. In addition, comparing the results with AV only case, second scheme increases the improvement obtained by AV individually in all cases. In the third scheme, combining CV and AV with a single control variate yields inferior results with regards to AV. However, as stated before, applying CV to all service times with AV produce very substantial increments in the improvement.

4.3.2. Latin Hypercube Sampling + Control Variates (LHS+CV)

In this case, we apply LHS to all service times at k=2,3 and consider using different control variates during the application of CV. The same combination scheme is applied again as in the M/M/1 case. In the use of single control variates, the degrees of freedom is taken as 28 and 18 at k=2 and k=3 stratification levels while in the ‘ALL’ case, 24 and 14 are used for the degrees of freedoms. The results are given in Tables 4.23 - 4.25.

		LHS, with k=2 + CV					
	No	Service1	Service 2	Service3	Service4	Service5	ALL
	1	6.78%	17.14%	12.50%	15.93%	13.99%	12.00%
	2	-4.53%	7.05%	-4.92%	-2.35%	-11.85%	-8.15%
	3	10.18%	7.33%	9.91%	11.64%	11.88%	12.08%
	4	-12.82%	-10.40%	-6.55%	6.95%	-7.43%	11.86%
	5	15.28%	12.28%	10.21%	15.20%	2.66%	14.61%
	6	15.13%	41.78%	25.81%	15.48%	16.05%	38.37%
	7	11.55%	27.48%	25.09%	11.28%	17.55%	20.60%
	8	15.52%	15.12%	15.59%	26.26%	12.92%	25.99%
	9	32.51%	30.62%	33.92%	37.57%	29.44%	35.25%
	10	20.16%	19.81%	21.13%	24.07%	19.05%	27.85%
Confidence Intervals for Different Control Variates	Average	**10.98%**	**16.82%**	**14.27%**	**16.20%**	**10.43%**	**19.05%**
	Std. Dev.	3.99%	4.58%	4.12%	3.49%	3.96%	4.31%
	Upper Limit	19.98%	27.17%	23.57%	24.09%	19.39%	28.79%
	Lower Limit	1.97%	6.47%	4.96%	8.32%	1.47%	9.30%
	Half-length	9.01%	10.35%	9.31%	7.89%	8.96%	9.75%

Table 4.23 *Results of CV and LHS with k=2 and Confidence Intervals for Improvements in Half-length for different Control Variates*

	LHS, with k=3 + CV					
No	Service1	Service 2	Service3	Service4	Service5	ALL
1	2.56%	2.72%	6.44%	9.24%	1.05%	29.43%
2	6.81%	7.64%	14.75%	25.04%	18.78%	17.65%
3	-18.79%	-15.50%	-11.95%	-4.76%	1.26%	1.71%
4	12.78%	28.18%	21.02%	2.79%	3.55%	34.88%
5	-5.53%	0.63%	-5.68%	-0.11%	2.01%	6.99%
6	13.69%	15.07%	33.45%	14.84%	8.31%	1.54%
7	34.81%	31.50%	37.01%	31.91%	35.17%	42.22%
8	27.17%	23.07%	18.11%	24.06%	31.61%	36.90%
9	17.07%	20.69%	21.75%	19.20%	18.84%	21.66%
10	16.71%	20.80%	14.09%	13.94%	13.55%	25.89%

Confidence Intervals for Different Control Variates	Average	**10.73%**	**13.48%**	**14.90%**	**13.61%**	**13.41%**	**21.89%**
	Std. Dev.	4.90%	4.59%	4.88%	3.76%	3.97%	4.65%
	Upper Limit	21.79%	23.85%	25.93%	22.10%	22.38%	32.38%
	Lower Limit	-0.34%	3.11%	3.87%	5.13%	4.45%	11.39%
	Half-length	11.06%	10.37%	11.03%	8.49%	8.97%	10.50%

Table 4.24 *Results of CV and LHS with k=3 and Confidence Intervals for Improvements in Half-length for different Control Variates*

	Confidence Intervals for Improvements in Standard Deviation											
	LHS with k=2 + CV						LHS with k=3 + CV					
	Serv. 1	Serv. 2	Serv. 3	Serv. 4	Serv. 5	ALL	Serv. 1	Serv. 2	Serv. 3	Serv. 4	Serv. 5	ALL
Average	**13.02%**	**18.73%**	**16.24%**	**18.13%**	**12.48%**	**21.52%**	**14.94%**	**17.56%**	**18.91%**	**17.69%**	**17.49%**	**27.13%**
Std. Dev.	3.89%	4.47%	4.02%	3.41%	3.87%	4.18%	4.66%	4.37%	4.65%	3.58%	3.78%	4.33%
Upper Limit	21.82%	28.84%	25.33%	25.83%	21.24%	30.97%	25.48%	27.44%	29.42%	25.77%	26.04%	36.92%
Lower Limit	4.22%	8.62%	7.14%	10.42%	3.73%	12.07%	4.39%	7.68%	8.40%	9.60%	8.95%	17.34%
Half-length	8.80%	10.11%	9.09%	7.71%	8.75%	9.45%	10.54%	9.88%	10.51%	8.09%	8.54%	9.79%

Table 4.25 *Confidence Intervals for Improvements in Standard Deviation by LHS with k=2,3 and CV when different service times are taken as Control Variates*

When CV is applied to all service times, increasing the stratification level increases the average improvement. Nonetheless, this does not hold for the application of CV to individual service times. Since all lower limits are greater than zero, this combination should provide an improvement in the precision. Interestingly, in the second case of k=2, a negative improvement is observed in ‘ALL’ case. In addition, as seen in the sixth case of k=3 case, this combination can produce poorer improvements when CV is applied to all service times with respect to the application of CV to any of the service times individually.

Average improvements indicate that applying CV with LHS produces worse results than CV alone when all service times are used as control variates. However, based on the combinations of CV with AV or LHS, we conclude that any combination produces better results in single control variate cases while it gives inferior improvements when all service times are used as control variates. We note that applying CV further provides more improvement for each of ten cases regardless of the control variate.

4.3.3. Overview of the Results

We summarize the overall results in Tables 4.26 and 4.27. These are itemized as follows:

Method		Serial Line Production System					
		Half Length			Standard Deviation		
		Improv. (Half L.)	Upper Limit	Lower Limit	Improv. (Std. D.)	Upper Limit	Lower Limit
AV-ALL		7.61%	17.24%	-2.02%	9.60%	19.02%	0.17%
LHS, k=2		6.59%	16.38%	-3.21%	8.60%	18.18%	-0.98%
LHS, k=3		6.97%	16.81%	-2.87%	10.93%	20.35%	1.51%
CV-Service Time 1		4.38%	8.31%	0.45%	4.38%	8.31%	0.45%
CV-Service Time 2		5.89%	8.87%	2.91%	5.89%	8.87%	2.91%
CV-Service Time 3		8.99%	14.60%	3.37%	8.99%	14.60%	3.37%
CV-Service Time 4		7.84%	14.05%	1.64%	7.84%	14.05%	1.64%
CV-Service Time 5		4.98%	10.13%	-0.16%	4.98%	10.13%	-0.16%
CV-ALL		32.32%	36.23%	28.41%	32.46%	36.36%	28.57%
PS- Service Time 3	s=2	6.01%	11.43%	0.60%	7.17%	12.52%	1.82%
	s=3	6.33%	12.15%	0.50%	7.54%	13.27%	1.81%
	s=4	7.17%	12.80%	1.53%	8.39%	13.99%	2.78%
	s=5	7.05%	12.11%	1.98%	8.34%	13.37%	3.31%

Table 4.26 *Overall Improvements in Half-length and Standard Deviation by Each Technique alone*

Method		Serial Line Production System					
		Half-length			Standard Deviation		
		Improv. (Half L.)	Upper Limit	Lower Limit	Improv. (Std. D.)	Upper Limit	Lower Limit
AV + CV-ALL		23.58%	34.05%	13.11%	25.91%	36.06%	15.76%
LHS + CV-ALL	k=2	19.05%	28.79%	9.30%	21.52%	30.97%	12.07%
	k=3	21.89%	32.38%	11.39%	27.13%	36.92%	17.34%

Table 4.27 *Overall Improvements of Combinations in Half-length and Standard Deviation*

- In this case, 'CV-ALL' produces the best improvement even it is compared to the combined applications. Thus the use of methods in combination may not always give better results. As seen, integrating CV-ALL with any other technique produces worse results.
- Inconsistently with the M/M/1 system, the combination of AV and CV performs worse than the stand-alone application of CV. We explain this using some intuitive results. Firstly, the total gain in the combination of AV and CV is equal to the gain obtained with the stand-alone application of AV plus the additional gain obtained with the application of CV. This is expressed in the following equation:

$$\text{Total Gain of } AV+CV = Gain(AV) + Gain(CV|AV)$$

We believe that *Gain(CV)>Gain(CV|AV)*. The combined use will perform better than the stand alone use of CV if *Gain(AV)>Gain(CV)-Gain(CV|AV)*. In the M/M/1 system, *Gain(CV|AV)* were 24.82%-14.03%=10.79% and 34.81%-15.45%=19.36% for 0.9 and 0.5 utilizations, respectively. Hence, *Gain(CV)*=16.95%>10.79% for 0.9 and *Gain(CV)*=31.97%>19.36 for 0.5 utilizations. Thus the better performance of the combination is expected. Nevertheless, in the serial line case, *Gain(CV|AV)*=32.32%-7.61%=24.71% is larger than *Gain(AV)*=7.61%. Therefore, an additional improvement with AV and CV with respect to CV alone cannot be expected.

- The combined applications of AV and LHS with CV perform better than their stand-alone applications. Since the lower limits of the confidence intervals for the average improvements in half-lengths are larger than zero, these combined uses are expected to provide improvements in the half-length and standard deviation.
- In general, applying CV or PS after inducing correlation among the replications through AV or LHS increases the improvements attained by AV or LHS individually.
- Also, applying CV to negatively correlated data produces better results than the independent case with one exception; when all service times are used as control variates, applying CV to independent data resulted in better improvement.
- AV and CV produce the best result in half-length while LHS with k=3 and CV produce the best result in the standard deviation. Not only the selection of the performance measure but also the best combination depends on the purpose of the study.
- As the stratification level increases, the performance of LHS seems to increase, however, this is not statistically significant.
- The combination of CV with AV performs better than the combination with LHS.

Chapter 5

Analysis of VRTs in the Output Analysis of an Inventory System under (s,S) Policy

5.1. (s,S) Inventory Policy

In this chapter, another commonly used inventory system with (s,S) Inventory Policy will be studied by the applications of VRTs. In this inventory policy, demands occur continuously over time and when the inventory position falls below the reorder point s, a new order in the amount that carries the inventory position to maximum inventory level S is placed. Inventory position equals to the inventory on hand plus the outstanding orders minus backlogged demand.

In these systems, there are usually three types of costs involved. These are holding cost that occurs whenever the inventory on hand is greater than zero, backlogging cost that occurs whenever the on hand inventory is negative, and ordering cost that is the fixed cost when an order is placed. In our applications of VRTs to inventory systems, however, we will not deal with the costs but instead focus directly on the amounts of either average inventory on hand or the average backlogged amount. More explicitly, we will try to reduce the variances of the estimator of average inventory on hand (IOH).

In our case, we will assume weekly demands having a normal distribution with mean 19.23 and standard deviation 5.658. Also, the orders will arrive at the beginning of each week with a constant lead time of two weeks. Maximum inventory level and the reorder point are determined to be 150 and 20 respectively. Therefore whenever the inventory on hand drops under 20 in week 7, say 15, an order of 150-15=135 is placed. Then, this order is expected to arrive at the beginning of week 9.

Inventory on hand is taken to be our performance measure in this inventory system. During this application, the individual and combined VRTs presented in Table 5.1 are applied to reduce the variation around the inventory on hand estimate under this policy:

(s,S) Inventory Policy	
AV	PS (s=2, 3, 4, 5)
LHS (k=2, 3)	AV+CV
CV	LHS (k=2, 3) +CV

Table 5.1 *List of Single and Integrated VRTs that will be Applied to (s,S) Inventory Model*

In order to determine the warm up period, a single simulation run has been taken and the graph of on hand inventory versus the number of weeks is plotted in Figure 5.1:

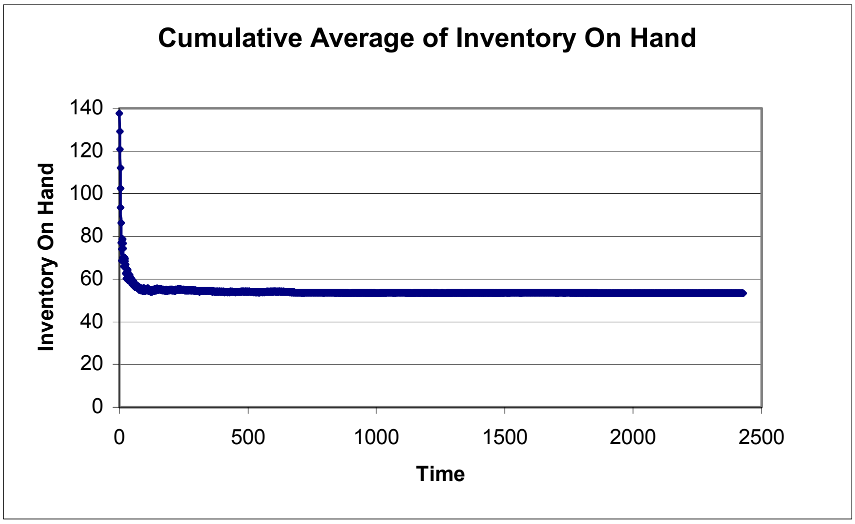

Figure 5.1 *Warm up Period for the (s,S) Inventory Model*

Hence, the warm up period is estimated to be around the first 200 weeks. However, the number of backlogged items reaches steady state around the week 500, which is not presented here. Thus to be on the safe side, we discard the first 800 weeks and calculate statistics for the next 52,000 weeks that corresponds to 1,000 years of the system operation in the steady state. By this way, we assure the convergence of statistics to the steady state. Again, we perform experimentation with 60 replications in each case with the same reasoning stated in the previous chapters. We construct confidence intervals for the improvements in half-length and standard deviation via 10 experiments during all applications of VRTs alone or in combination. Then, the results of the 10 experiments are used during the performance evaluation.

5.2. Application of VRTs Individually

5.2.1. Independent Case

In this inventory model only demand exists as a random input variable since the lead-time is assumed to be constant. Thus in each of 10 cases different demands are used. We present the results of 10 experiments together with the stream numbers and the initial seeds in Table 5.2.

Independent Case					
No	Seed	Mean	Std. Dev.	CV	Half-length
1	1	53.762413	0.008889	0.000165	0.017787
2	2	53.762056	0.007699	0.000143	0.015406
3	3	53.767917	0.006834	0.000127	0.013674
4	4	53.771916	0.008081	0.000150	0.016169
5	5	53.767663	0.007432	0.000138	0.014872
6	6	53.765261	0.007969	0.000148	0.015945
7	7	53.761808	0.008139	0.000151	0.016285
8	8	53.764855	0.009038	0.000168	0.018085
9	1-13547	53.748947	0.009232	0.000172	0.018472
10	2-10325	53.758820	0.009012	0.000168	0.018032

Table 5.2 *Results of the Ten Independent Replications with the Stream and Initial Seed Numbers*

According to the results of ten experiments, benchmark values for half-length and standard deviation are presented in Table 5.3.

	Estimated Half-length and Standard Deviation				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
Half-length	**0.016473**	0.000502	0.016832	0.016114	0.000359
Std. Dev.	**0.008232**	0.000251	0.008412	0.008053	0.000179

Table 5.3 *Benchmark Half-length and Standard Deviation Values*

5.2.2. Antithetic Variates (AV)

Only input to this model is the demand generated for each week. Therefore, we induce negative correlation among the demand values and observe the results in the IOH value. We use the same streams and initial seeds as in the independent case. The results are given in Tables 5.4 and 5.5.

The results indicate that AV provides improvements of 10.70% and 12.62% in the half-length and the standard deviation, respectively. Since the confidence intervals do not include zero, we expect that AV increases the precision in the (s,S) inventory model. Conversely, there exist three negative improvements in the half-length and standard deviation among these 10 cases. In order to make a better conclusion, we also examine the results of the comparison with the corresponding independent run based on the reasoning behind the VRTs. That is, after taking 30 runs we decide on whether to generate 30 additional runs independently of the previous runs or to construct the complementary runs of the previous 30 runs through the induction of negative correlation. In Tables 5.6 and 5.7, we present the results of this comparison in terms of half-lengths and standard deviation and the related confidence intervals for the average improvement.

AV								
No	Std. Dev.	Covariance	Correlation	Var(ODD)	Var(EVEN)	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	0.006769	-0.001429	-0.362446	0.005561	0.002795	0.013842	**15.97%**	**17.78%**
2	0.008316	0.000087	0.021459	0.004343	0.003782	0.017006	**-3.23%**	**-1.01%**
3	0.005826	-0.00108	-0.34865	0.002767	0.003466	0.011914	**27.67%**	**29.23%**
4	0.005551	-0.001211	-0.397778	0.003371	0.002748	0.011352	**31.09%**	**32.57%**
5	0.007431	-0.000897	-0.213165	0.004332	0.00409	0.015197	**7.74%**	**9.73%**
6	0.00867	-0.000701	-0.135812	0.004498	0.005925	0.017731	**-7.63%**	**-5.32%**
7	0.007214	-0.000802	-0.206684	0.004508	0.003341	0.014752	**10.45%**	**12.37%**
8	0.007328	-0.001383	-0.300313	0.004616	0.004592	0.014985	**9.03%**	**10.99%**
9	0.008237	-0.001185	-0.229894	0.006291	0.004221	0.016846	**-2.26%**	**-0.06%**
10	0.006595	-0.00168	-0.406287	0.005431	0.003146	0.013486	**18.13%**	**19.90%**

Table 5.4 *Results of AV*

	Confidence Intervals				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
Half-length	**10.70%**	4.08%	19.91%	1.48%	9.22%
Std. Dev.	**12.62%**	3.99%	21.64%	3.60%	9.02%

Table 5.5 *Confidence Intervals for Improvements by AV*

No	Half-length			Standard Deviation		
	Independent	AV	Improv.	Independent	AV	Improv.
1	0.017787	0.013842	22.18%	0.008889	0.006769	23.85%
2	0.015406	0.017006	-10.38%	0.007699	0.008316	-8.01%
3	0.013674	0.011914	12.87%	0.006834	0.005826	14.75%
4	0.016169	0.011352	29.79%	0.008081	0.005551	31.30%
5	0.014872	0.015197	-2.19%	0.007432	0.007431	0.01%
6	0.015945	0.017731	-11.20%	0.007969	0.008670	-8.81%
7	0.016285	0.014752	9.41%	0.008139	0.007214	11.36%
8	0.018085	0.014985	17.14%	0.009038	0.007328	18.93%
9	0.018472	0.016846	8.81%	0.009232	0.008237	10.77%
10	0.018032	0.013486	25.21%	0.009012	0.006595	26.82%

Table 5.6 *Individual Improvements in Half-length and Standard Deviation by AV*

	Confidence Intervals				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
Half-length	**10.17%**	4.53%	20.40%	-0.07%	10.23%
Std. Dev.	**12.10%**	4.43%	22.11%	2.08%	10.01%

Table 5.7 *Confidence Intervals for Improvements in Half-length and Standard Deviation by AV*

As a result, AV cannot promise an improvement in half-length or precision. Even though the negative correlation is induced among the replications in nine cases, the inferior results are observed as well. In fact, IOH is monotone non-increasing function of the weekly demand. We can establish the reasoning as follows: As the weekly demand increases, the probability of backlogging increases and since there exists an inverse correlation with the IOH and backlogged demands, IOH should decrease and vice versa. Also, increasing demand sometimes may not decrease the IOH.

5.2.3. Latin Hypercube Sampling (LHS)

LHS is applied to stratify only the demand values with k=2 and k=3. We consider 30 and 20 macro replications for each stratification level, respectively. The streams and initial seeds given in Table 5.8 are used to generate permutations and service times for both k=2 and k=3. Results are presented in Tables 5.9 - 5.11 for both stratification levels.

No	1	2	3	4	5	6	7	8	9	10
Permutation	1	3	5	7	1-23251	3-19631	5-31235	7-23157	1-23591	3-11233
Demand	2	4	6	8	2-13571	4-17913	6-12357	8-11537	2-15423	4-21357

Table 5.8 *Stream and Initial Seed Numbers for LHS*

	LHS with k=2					LHS with k=3				
No	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)	Mean	Std.Dev.	Half-length	Improv. (Half L.)	Improv. (Std. D.)
1	53.776662	0.006897	0.014104	14.38%	16.23%	53.761735	0.010302	0.021530	-30.70%	-25.13%
2	53.765563	0.008420	0.017218	-4.52%	-2.28%	53.775555	0.006216	0.012992	21.13%	24.49%
3	53.779064	0.007350	0.015030	8.76%	10.72%	53.775541	0.007209	0.015068	8.53%	12.43%
4	53.774270	0.006914	0.014140	14.17%	16.01%	53.761908	0.007179	0.015005	8.91%	12.79%
5	53.773020	0.007773	0.015896	3.50%	5.58%	53.770918	0.006418	0.013414	18.57%	22.04%
6	53.764877	0.009012	0.018430	-11.88%	-9.47%	53.760900	0.006563	0.013716	16.74%	20.28%
7	53.769693	0.006605	0.013507	18.00%	19.77%	53.753445	0.007236	0.015123	8.20%	12.10%
8	53.752636	0.006039	0.012350	25.03%	26.64%	53.761962	0.007753	0.016204	1.63%	5.82%
9	53.758533	0.009229	0.018873	-14.57%	-12.10%	53.774225	0.005075	0.010606	35.62%	38.36%
10	53.769076	0.007390	0.015113	8.26%	10.23%	53.780814	0.005242	0.010957	33.49%	36.32%

Table 5.9 *Results of LHS with k=2 and k=3*

	Half-length				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
LHS, k=2	**6.11%**	4.10%	15.39%	-3.16%	9.27%
LHS, k=3	**12.21%**	5.90%	25.55%	-1.13%	13.34%

Table 5.10 *Confidence Intervals for Improvements in Half-length by LHS with k=2 and k=3*

	Standard Deviation				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
LHS, k=2	**8.13%**	4.02%	17.21%	-0.94%	9.07%
LHS, k=3	**15.95%**	5.65%	28.72%	3.18%	12.77%

Table 5.11 *Confidence Intervals for Improvements in Standard Deviation by LHS with k=2 and k=3*

The results indicate that increasing the stratification level contributes much to the resulting improvement both in half-length and the standard deviation. However, both confidence intervals include zero implying that LHS may not provide any improvement in half-length and even produce inferior results. This inference is also applicable to standard deviation at the k=2. Only at the k=3, LHS is likely to guarantee a positive improvement in the standard deviation.

Having summarized the results of LHS and AV in Table 5.12, we see that AV performs better than LHS with k=2 in terms of both half-length and standard deviation. However, LHS with k=3 results in better improvements compared to AV although AV produces larger lower limits than LHS with k=3. In fact, LHS with k=3 have very large half-lengths for the average improvements. Therefore, even though AV results in less improvement on the average, it provides positive improvements more consistently than LHS with k=3.

		Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
Half-length	AV	**10.70%**	4.08%	19.91%	1.48%	9.22%
	LHS, k=2	**6.11%**	4.10%	15.39%	-3.16%	9.27%
	LHS, k=3	**12.21%**	5.90%	25.55%	-1.13%	13.34%
Standard Deviation	AV	**12.62%**	3.99%	21.64%	3.60%	9.02%
	LHS, k=2	**8.13%**	4.02%	17.21%	-0.94%	9.07%
	LHS, k=3	**15.95%**	5.65%	28.72%	3.18%	12.77%

Table 5.12 *Comparison of AV and LHS in terms of Half-length and Standard Deviation Improvement*

5.2.4. Control Variates (CV)

Weekly demand that is the single input to the system is used as the control variate. Even though a non-linear relationship is anticipated between the demand and the IOH, this application demonstrates whether it can be estimated with a linear relationship. Theoretically known mean of weekly demand will be taken as 19.23. During the implementation of CV we use t value with degrees of freedom 58. In Tables 5.13 and 5.14, the results are presented.

	CV					
No	Mean	Std. Dev.	Correlation	Half-length	Improv. (Half L.)	Improv. (Std.D.)
1	53.765910	0.007746	-0.490607	0.015507	5.86%	5.91%
2	53.762491	0.007342	-0.301089	0.014699	10.77%	10.81%
3	53.764108	0.006011	-0.475718	0.012034	26.95%	26.98%
4	53.768870	0.006962	-0.507727	0.013937	15.39%	15.44%
5	53.764896	0.006920	-0.364848	0.013854	15.90%	15.94%
6	53.762167	0.007148	-0.441954	0.014310	13.13%	13.17%
7	53.760526	0.007017	-0.506517	0.014049	14.72%	14.76%
8	53.765374	0.007996	-0.466136	0.016008	2.82%	2.87%
9	53.758152	0.007740	-0.544958	0.015496	5.93%	5.98%
10	53.758774	0.007321	-0.583106	0.014657	11.03%	11.07%

Table 5.13 *Results of CV*

	Confidence Intervals				
	Average	Std.Dev.	Upper Limit	Lower Limit	Half-length
Half-length	**12.25%**	2.16%	13.79%	10.71%	1.54%
Std. Dev.	**12.29%**	2.16%	13.84%	10.75%	1.54%

Table 5.14 *Confidence Intervals for Improvements in Half-length and Standard Deviation*

According to the results given in Table 5.14, CV consistently provides a considerable improvement in both the half-length and the standard deviation. The difference between the average improvements belonging to half-length and standard deviation occurs due to the t value. Observing the individual improvement values, we observe that all improvement values are positive and thus the performance of the CV is excellent. As a result, CV provides an improvement between 10.71% and 13.79% in the half-length with 95% probability.

A comparison of the CV with the previously discussed two methods, AV and LHS, reveals that CV provides much more consistency and larger lower limits in the improvement levels in both half-lengths and standard deviation. However, neither LHS nor AV has such a high lower limit even though their averages are greater in some cases. Thus, the gain with CV is more likely than LHS or AV, which means that applying CV could be more risk efficient.

5.2.5. Poststratified Sampling (PS) (s=2,3,4,5)

In this case, backlogged demand and the weekly demand turn out to be the possible stratification variates. Both of these variables have a negative correlation with the IOH. Intuitively, weekly demand has a more correlation with the IOH since it directly affects the IOH whereas backlogged demand is again a consequence of demand. This is also confirmed by our experimental results. The correlation between the weekly demand and the IOH is –0.468266 based on the ten estimates while it is –0.278161 in the backlogged demand case. Therefore, we choose the weekly demand to stratify the IOH values, where a non-linear relationship exists between the weekly demand and IOH.

We consider four different stratification levels s=2,3,4,5 and form each stratum by calculating the intervals of equal probability and the optimal allocation schemes. The results are presented in Tables 5.15 and 5.16.

The results indicate that the equal probability scheme is dominated by the other scheme even if the differences can be considered negligible. Examining the results of the two schemes, it is seen that the optimal allocation scheme produced the best result 10.19% at s=5. Confidence intervals of the average improvements include the zero in both s=2 levels regardless of the stratification scheme. However, lower limits are larger than zero with higher stratification levels. This means that PS is expected to provide improvements in both the half-length and standard deviation. Once again, this improvement is not guaranteed. The optimal allocation scheme requires the stratification variate to be normally distributed and since the average demand in this case is normal due to the Central Limit Theorem, we will prefer the usage of this scheme in single and combined applications.

Improvement Levels in Half Length									
No	Optimal Allocation Scheme				No	Equal Probability Scheme			
	s=2	s=3	s=4	s=5		s=2	s=3	s=4	s=5
1	-0.18%	-0.71%	1.24%	5.85%	**1**	-0.18%	2.94%	-0.01%	-0.84%
2	8.27%	10.37%	10.11%	8.78%	**2**	8.27%	7.59%	8.77%	9.16%
3	24.85%	24.06%	24.51%	24.50%	**3**	24.85%	27.71%	25.45%	23.77%
4	7.59%	9.86%	17.26%	15.40%	**4**	7.59%	9.27%	9.18%	12.29%
5	13.96%	10.26%	12.75%	9.68%	**5**	13.96%	11.62%	12.74%	14.64%
6	12.26%	8.62%	16.12%	10.82%	**6**	12.26%	9.49%	11.48%	11.90%
7	11.04%	11.47%	10.48%	10.30%	**7**	11.04%	11.99%	12.80%	12.13%
8	-7.05%	-2.66%	-5.94%	0.75%	**8**	-7.05%	-2.35%	-3.79%	-5.28%
9	-7.99%	2.50%	6.43%	3.32%	**9**	-7.99%	0.62%	4.04%	3.62%
10	-0.97%	8.73%	3.08%	12.56%	**10**	-0.97%	7.96%	9.69%	7.58%

Confidence Intervals for Different Stratification Levels		OA s=2	OA s=3	OA s=4	OA s=5	EP s=2	EP s=3	EP s=4	EP s=5
	Average	**6.18%**	**8.25%**	**9.60%**	**10.19%**	**6.18%**	**8.68%**	**9.04%**	**8.90%**
	Std. Dev.	3.24%	2.37%	2.79%	2.10%	3.24%	2.59%	2.53%	2.60%
	Upper Limit	13.49%	13.60%	15.90%	14.93%	13.49%	14.54%	14.75%	14.78%
	Lower Limit	-1.14%	2.90%	3.31%	5.46%	-1.14%	2.83%	3.32%	3.01%
	Half Length	7.32%	5.35%	6.29%	4.74%	7.32%	5.86%	5.71%	5.89%

Table 5.15 *Results of PS and Confidence Intervals for Improvements at each Stratification Level obtained with Optimal Allocation and Equal Probability Schemes*

	Confidence Intervals for Improvements in Standard Deviation							
	Optimal Allocation Scheme				Equal Probability Scheme			
	s=2	s=3	s=4	s=5	s=2	s=3	s=4	s=5
Average	**6.19%**	**8.35%**	**9.74%**	**10.41%**	**6.19%**	**8.71%**	**9.17%**	**9.16%**
Std. Dev.	3.23%	2.33%	2.73%	2.07%	3.23%	2.57%	2.47%	2.55%
Upper Limit	13.50%	13.61%	15.91%	15.08%	13.50%	14.53%	14.76%	14.93%
Lower Limit	-1.13%	3.09%	3.57%	5.73%	-1.13%	2.90%	3.58%	3.40%
Half Length	7.31%	5.26%	6.17%	4.67%	7.31%	5.81%	5.59%	5.77%

Table 5.16 *Confidence Intervals for Improvements in Standard Deviation by PS at each Stratification Level obtained with Optimal Allocation and Equal Probability Schemes*

In both half-length and standard deviation cases, there exists an increasing trend in the improvement as the stratification level increases. Similarly to the previous models, we see the superiority of CV over PS both in average improvements when the same random variable is used as control and stratification variates. Also, PS can result in positive improvements ranging on a larger spectrum than CV.

5.2.6. Overview of the Results

We summarize the overall results of the applications of each VRT individually Table 5.17:

		(s,S) Inventory Policy					
		Half-length			Standard Deviation		
Method		Improv. (Half L.)	Upper Limit	Lower Limit	Improv. (Std. D.)	Upper Limit	Lower Limit
AV		10.70%	19.91%	1.48%	12.62%	21.64%	3.60%
LHS, k=2		6.11%	15.39%	-3.16%	8.13%	17.21%	-0.94%
LHS, k=3		12.21%	25.55%	-1.13%	15.95%	28.72%	3.18%
CV		12.25%	13.79%	10.71%	12.29%	13.84%	10.75%
PS - Demand	s=2	6.18%	13.49%	-1.14%	6.19%	13.50%	-1.13%
	s=3	8.25%	13.60%	2.90%	8.35%	13.61%	3.09%
	s=4	9.60%	15.90%	3.31%	9.74%	15.91%	3.57%
	s=5	10.19%	14.93%	5.46%	10.41%	15.08%	5.73%

Table 5.17 *Overall Improvements in Half-length and Standard Deviation by Each Technique alone*

- In terms of the average improvement in the half-length, CV outperforms the other methods. This is followed by AV, whose improvement is also significant. The performance of PS is also very competitive; it yields significant and consistent (the improvement shows little variation from experiment to experiment) reductions. Only exception is observed at PS at s=2.
- Even though the mean improvement of LHS is the second, it displays highly volatile behavior in terms of the amount of improvement in the half-length. Specifically, the average improvements vary over a very large range, i.e. between -1.13% and 25.55%. Thus we rank this method last.
- In terms of the improvement in the standard deviation, the results are quite mixed. With respect to the average improvement, LHS with k=3 is the best. However, the confidence interval constructed around this mean is too large; hence one cannot easily rely on the mean improvement suggested by LHS. CV is the second and this is followed by AV and PS. With respect to the lower limit of the average improvement, CV stands out as the best method. This is followed by PS at s=5, AV and LHS.
- The worst results are produced by LHS at k=2 for the half-length and by PS at s=2 for the standard deviation.
- The confidence intervals constructed for AV do not include zero. This means that AV can provide an improvement in the half-length or standard deviation.
- Increasing the stratification level of LHS and PS improve their performance.
- Out overall conclusion is the use of CV since it yields the best improvement consistently.

5.3. Application of VRTs in Combination

5.3.1. Antithetic Variates + Control Variates (AV+CV)

In the M/M/1 and serial line cases, three separate schemes to combine the AV and CV have been applied. An interesting situation occurs in this model in the second scheme, which firstly combines the negatively correlated replications before the application of CV. The weekly demand that is normally distributed is selected as the control variate. Since it is the only input random variable, negative correlation is induced among the weekly demands during the application of AV. Since normal distribution is a symmetric distribution, taking the average of corresponding odd and even runs, we always obtain the same theoretical mean, 19.23, for all pairs. Thus CV cannot be applied because the correction part in the CV formula will always be zero. An input random variable that has a symmetric distribution cannot be used as the control variate in the second combination scheme of AV and CV if AV is applied to that input variable.

Hence, we present the results for the first and third schemes. In the first scheme, the improvements in standard deviation and the associated confidence interval are presented as in earlier cases. In the third case, the results are presented in terms of both half-length and standard deviation. Degrees of freedom is taken as 54 in the third scheme. The results are presented Tables 5.18 - 5.21.

	Individual Improvements in the Standard Deviation for the First Scheme									
No	1	2	3	4	5	6	7	8	9	10
Std. Dev.	20.07%	3.08%	32.08%	33.65%	12.41%	-5.23%	12.93%	14.50%	2.38%	21.27%

Table 5.18 *Individual Improvements in Standard Deviation for the First Combination Scheme*

	Confidence Interval for the First Scheme				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
Std. Dev.	**14.71%**	3.98%	23.71%	5.72%	8.99%

Table 5.19 *Confidence Interval for the Improvements in Standard Deviation by the First Scheme*

	Individual Improvements for the Third Scheme									
No	1	2	3	4	5	6	7	8	9	10
Half L.	10.73%	4.45%	21.80%	17.42%	17.09%	3.51%	19.18%	17.87%	5.42%	17.13%
Std. Dev.	10.77%	4.50%	21.84%	17.46%	17.13%	3.56%	19.22%	17.91%	5.47%	17.17%

Table 5.20 *Individual Improvements in Half-length and Standard Deviation for the Third Scheme*

	Confidence Interval for the Third Scheme				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
Half L.	**13.46%**	2.15%	18.32%	8.60%	4.86%
Std. Dev.	**13.50%**	2.15%	18.36%	8.65%	4.86%

Table 5.21 *Confidence Intervals for the Improvements in Half-length and Standard Deviation by the Third Combination Scheme*

These results indicate that both schemes are successful in the improvement of the standard deviation since all lower limits are very far from the zero. Lower limit for the standard deviation improvement is considerably larger in the third combination scheme compared to the first scheme but first has a larger average value than the third. However, according to the length of the confidence intervals, third scheme provides more consistent improvements. In addition, third scheme performed very well in half-length improvement.

Comparing the results with the AV and CV applied individually, it is obvious that both schemes contribute to the average improvement in the standard deviation. This also holds for the third scheme on the half-length improvement. Even though the lower limits for the combination are higher as compared to AV individually, these two schemes produce smaller lower limits than CV individually. This means that applying CV to negatively correlated runs can produce better results on the average but it increases the probability of the inferior results.

5.3.2. Latin Hypercube Sampling + Control Variates (LHS+CV)

The single input variable, demand, is assigned to be the control variate and stratified on k=2 and k=3 levels. Again, CV is directly applied to the macro replications obtained by using LHS (i.e., after taking the average of k micro replications), as if they are generated from independent runs. The results are presented Tables 5.22-5.25.

	Improvements of LHS with k=2 + CV									
No	1	2	3	4	5	6	7	8	9	10
Half L.	17.76%	18.76%	16.77%	15.47%	30.34%	-1.40%	20.60%	28.98%	-9.42%	13.02%
Std. Dev.	19.64%	20.62%	18.68%	17.41%	31.94%	0.93%	22.42%	30.61%	-6.91%	15.02%

Table 5.22 *Results of CV and LHS with k=2 for Improvements in Half-length and Standard Deviation*

	Improvements of LHS with k=3 + CV									
No	1	2	3	4	5	6	7	8	9	10
Half L.	20.62%	21.60%	12.65%	8.64%	23.05%	32.88%	13.06%	28.21%	38.56%	36.98%
Std. Dev.	24.37%	25.30%	16.77%	12.95%	26.68%	36.04%	17.16%	31.59%	41.46%	39.95%

Table 5.23 *Results of CV and LHS with k=3 for Improvements in Half-length and Standard Deviation*

	Confidence Interval for LHS with k=2 + CV				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
Half L.	**15.09%**	3.88%	23.86%	6.31%	8.77%
Std. Dev.	**17.04%**	3.79%	25.61%	8.46%	8.57%

Table 5.24 *Confidence Intervals for Half-length and Standard Deviation Improvements by CV+LHS, k=2*

	Confidence Interval for LHS with k=3 + CV				
	Average	Std. Dev.	Upper Limit	Lower Limit	Half-length
Half L.	**23.63%**	3.30%	31.08%	16.18%	7.45%
Std. Dev.	**27.23%**	3.14%	34.33%	20.13%	7.10%

Table 5.25 *Confidence Intervals for Half-length and Standard Deviation Improvements by CV+LHS, k=3*

According to the results, this combination appears to be very effective in reducing the variance and especially the results at k=3 are commendable. We also observe that increasing the stratification level of LHS improves the performance drastically in terms of the average improvements, lower limits, and half-lengths.

5.3.3. Overview of the Results

We summarize the overall results obtained through the stand-alone and combined applications of VRTs in Tables 5.26 and 5.27.

		(s,S) Inventory Policy					
		Half-length			Standard Deviation		
Method		Improv. (Half L.)	Upper Limit	Lower Limit	Improv. (Std. D.)	Upper Limit	Lower Limit
AV		10.70%	19.91%	1.48%	12.62%	21.64%	3.60%
LHS, k=2		6.11%	15.39%	-3.16%	8.13%	17.21%	-0.94%
LHS, k=3		12.21%	25.55%	-1.13%	15.95%	28.72%	3.18%
CV		12.25%	13.79%	10.71%	12.29%	13.84%	10.75%
PS - Demand	s=2	6.18%	13.49%	-1.14%	6.19%	13.50%	-1.13%
	s=3	8.25%	13.60%	2.90%	8.35%	13.61%	3.09%
	s=4	9.60%	15.90%	3.31%	9.74%	15.91%	3.57%
	s=5	10.19%	14.93%	5.46%	10.41%	15.08%	5.73%

Table 5.26 *Overall Improvements in Half-length and Standard Deviation by Each Technique alone*

		(s,S) Inventory Policy					
		Half-length			Standard Deviation		
Method		Improv. (Half L)	Upper Limit	Lower Limit	Improv. (StdD.)	Upper Limit	Lower Limit
AV + CV		13.46%	18.32%	8.60%	13.50%	18.36%	8.65%
LHS + CV	k=2	15.09%	23.86%	6.31%	17.04%	25.61%	8.46%
	k=3	23.63%	31.08%	16.18%	27.23%	34.33%	20.13%

Table 5.27 *Overall Improvements of Combinations in Half-length and Standard Deviation*

- In all cases, combined use of VRTs performs on the average better than their stand-alone applications.
- Lower limits of the confidence intervals for the average improvements in half-lengths and standard deviations are always larger than zero. This means that the combined methods are expected to provide improvement, in general.
- As in the case of M/M/1, the combination of CV produces better results than the stand-alone application. Recall that in the serial line case, the performance of CV deteriorates with AV due to the reasons explained in section 4.3.3.
- LHS with k=3 and CV is the best in half-length and standard deviation improvement.

- LHS with k=2 and PS with s=2 is the worst in half-length and standard deviation improvement.
- As expected the performance of LHS improves in combination with CV and PS as the level of stratification increases.
- The combinations of CV with LHS produced better results than the combination with AV. Thus CV performs well with LHS than AV in the inventory model.
- The input random variables having a symmetric distribution cannot be used as either control or stratification variate in the second combination scheme of AV and CV.
- As a result, the methods using auxiliary variables (CV and PS) and correlation induction (AV and LHS) should be used together in order to enhance the results.

Chapter 6

Conclusion

In this thesis, we analyzed four Variance Reduction Techniques (VRTs): Antithetic Variates (AV), Latin Hypercube Sampling (LHS), Control Variates (CV), and Poststratified Sampling (PS). We experimented with these methods alone and in combination on three different systems at steady state. Firstly, the simple M/M/1 system was considered at two different utilization levels (0.5 and 0.9). After that, a serial line production system of five stations with limited buffers between the stations was taken into account. An (s,S) Inventory Policy was the last system on which VRTs were applied. In those three systems, we used the 'Time-in-system', 'Throughput', and 'Inventory on Hand' as the performance measures respectively. During the research, the precision and the variance of those estimates were intended to be improved.

According to the results obtained in each system, there is no guarantee of reduction in the variance or increase in the precision. This is confirmed by examining the individual improvement values in ten different cases for each method. The use of VRTs may even produce even worse results (i.e., backfires). The success of VRTs depends on both the system characteristics and the technique applied in order to improve the precision and the variance. Even though CV was the best in all three systems, the performances of the other methods differ in each system. In M/M/1 system at the low utilization, PS with s=4 and s=5 stratification levels produced very commendable results. These results were close to the improvement level achieved by the CV. In the high utilization level, LHS resulted in nearly the same improvement amount with the CV. In the (s,S) inventory policy, LHS with k=3 and AV yielded quite good improvements on the average.

PS performed well in the M/M/1 system at the low utilization since it allows the choice of a stratification variate that has a high correlation with the performance measure. In the other systems, the performance of PS was not good since there was not such a variate having a high correlation with the response variable. In the serial line production system, CV performed better than any other methods and combination since it allows the use of more than one variable as the control variate. In our implementations, we used service times for all five stations as the control variates since we knew the distribution function of each service times for each station.

Comparing PS and CV in these all cases, we observed the relative advantages of each method over the other. As long as you know the theoretical mean of a random variable, you can use it as the additional control variate. By this way, you can correct the raw performance measure, as many times as there are control variates. As stated, one requirement of CV is to select a control variate whose mean value is known beforehand. PS requires the choice of an input variable as the stratification variate, whose theoretical distribution is known. However, PS does not allow the use of more than one variable as the stratification variate. In the serial line case, the differences in the performances of CV and PS are totally based on the fact explained above.

On the other hand, the performances of CV and PS are comparable when the same variable is used as the control or stratification variate. In this case, CV outperformed PS in all three systems according to the average improvement values. This means that the use of the knowledge of a true mean value and correcting the response variable according to this mean should be preferred over the stratification provided by the same variate but using its distribution. This is the observed result according to individual improvement values. In some instances of improvements with PS produced better results than CV even if the same variable is chosen as the stratification and control variate.

In all three systems, CV or PS that do not alter the inputs of the simulation model gave the best improvement in the half-length. They also resulted in the best performance in the variance reduction in the M/M/1 at the low utilization and serial line cases. Nevertheless, in the (s,S) inventory policy, LHS with k=3 stratification level achieved the best reduction in the variance of the inventory on hand. In addition, LHS with k=2 resulted in best improvement in the variance in the M/M/1 system at the high utilization. As a result, the techniques, which try to extract more information using auxiliary or secondary variables perform better with respect to the methods inducing negative correlation especially for the half-length improvement. Interestingly, LHS with k=3 produced the best reduction in variance in the (s,S) inventory policy even though this is not reflected to the half-length improvement due to the loss in degrees of freedom. Therefore, among the individual methods, CV or PS should be preferred over AV and LHS and the choice of CV or PS should be determined according to the system characteristics.

In this manner, another observation regarding the implementation of CV and PS can be as follows. Since the analysis of output data through CV or PS requires a negligible time compared to the simulation run time in most practical cases, applying CV and PS at all possible cases and then selecting the best will be the most appropriate strategy for the variance reduction.

CV and PS require extra knowledge about the system. For example, the analysts should know the relationships between the time-in-system and the service time or waiting time in queue. A successful implementation of CV or PS is dependent on the availability of other measures whose distribution and mean are known theoretically. On the other hand, AV and LHS do not have such a requirement and thus no effort is spent on the analysis of the relationships between the variables of the system. As a result, if the analysis of the system is too difficult, then AV or LHS can be preferred over CV or PS in reducing the variance of the performance measure. Of course, the requirements of the AV or LHS such as monotone relationships should be satisfied. In fact, generating the random variables if possible via the inverse transform method meets the requirements of those two methods.

In general, using the methods in combination provides more improvement that the stand-alone applications. However, this improvement is not equal to the sum of the individual improvements. Of course, some exceptions exist to this case. This is again related with the system characteristics. For example, in the serial line production system, using any of AV or LHS with CV resulted in inferior improvement compared to the use of CV alone when all service times were taken as the control variates.

In combined applications, AV and CV resulted in best improvement levels in half-length and standard deviation at both utilization levels in the M/M/1 system. However, CV and LHS with k=3 produced close improvement levels to the best case. In the serial line production system, AV when applied to all service times and CV when all service times are taken as control variates provided best improvement in the half-length while LHS with k=3 and CV produced the best reduction in the variance. Again, we mention that CV alone without AV or LHS provided the best improvement both in the half-length and the standard deviation. In fact, AV and CV and LHS with k=3 and CV produced close results. In the (s,S) inventory model, CV when applied to negatively correlated data by LHS with k=3 performed the best in the improvement of both half-length and the standard deviation.

Observing the results of the stand-alone and combined application of VRTs, we can easily realize that VRTs provide more improvement in M/M/1 system compared to serial line and inventory cases on the average. Although, it seems intuitive that VRTs performs better in less complex systems, this is not valid since the performances of the VRTs directly depend on the desired performance measure and selected control or stratification variates. Even in a much complex system, better improvement levels can be obtained by the selection of good control or stratification variates.

Applicability of the some methods in combination depends on the input variables. For example, in the (s,S) inventory system, we could not apply the second combination scheme of AV and CV since the negative correlation is induced among the only input random variable that has a symmetric distribution. In this case, averaging the odd and even numbered replication averages, we obtained the theoretical mean of the input random variable for all pairs. As a result, especially in the combinations of AV, an input random variable having a symmetric distribution cannot be selected as neither the control variate nor the stratification variate if a negative correlation is induced among that input variables via AV.

Although we applied four methods to three different systems in this thesis, there is much need in a number of areas such as further applications of these four methods in some other systems and the integrated applications of other methods available in the literature. Nevertheless, extending the basic assumptions of VRTs will be a vital contribution to this subject. For example, using the estimated mean instead of the theoretical mean in CV would enhance the applicability of the CV to the output random variables as discussed in [31]. In addition, the two additional combination schemes available for AV and CV can be extended to LHS+CV and LHS+PS, however, the combined estimator should be proved to be unbiased. As shown theoretically, the third combination scheme of AV and CV was the best provided that the simple assumptions are satisfied. Again, this scheme can prove to be the best in the other three combinations as well. Finally, instead of combinations in two, studies can be extended to the combinations of three or more methods simultaneously.

References

1. Ahmed, M.A., Gross, D., and Miller, D.R. (1987), "Control Variate Models for Sensitivity Estimates of Repairable Item Systems", *Proceedings of the 1987 Winter Simulation Conference,* 324-333.

2. Avramidis, A.N. and J.R., Wilson. (1996), Integrated Variance Reduction Strategies for Simulation. *Operations Research,* **44**: 327-346.

3. Avramidis, A.N. and Wilson, J.R. (1993), "A Splitting Scheme for Control Variates", *Operations Research Letters,* **14**: 187-198.

4. Avramidis, A.N. and Wilson, J.R. (1996), "Correlation Induction Techniques for Estimating Quantiles in Simulation Experiments", *Operations Research,* **46**: 574-591.

5. Bauer, K.W., Venkatraman, S. and Wilson, J.R. (1987), "Estimation Procedures Based on Control Variates with Known Covariance Matrix", *Proceedings of the 1987 Winter Simulation Conference,* 334-341.

6. Burt, J.M, and Gaver, D.P. (1970), "Simple Stochastic Networks: Some Problems and Procedures", *Naval Research Logistics Quarterly,* **17**: 439-459.

7. Calvin, J. and Nakayama, M. (1997), "A New Variance Reduction Technique for Regenerative Simulations of Markov Chains", *Proceedings of the 1997 Winter Simulation Conference.*

8. Carson, J.S. and Law, A.W. (1980), "Conservation Equations and Variance Reduction in Queuing Simulations", *Operations Research,* **28**: 535-546.

9. Cheng, R.C.H. (1984), "Antithetic Variate Methods for Simulation Processes with Peaks and Troughs", *European Journal of Operations Research,* **15**: 227-236.

10. Glynn, P.W. and Whitt, W. (1989), "Indirect Estimation of $L=\lambda W$", *Operations Research,* **37**: 82-103.

11. Heidelberger, P. (1993), "Fast Simulation of Rare Events in Queuing and Reliability Models", *Performance Evaluation of Computer and Communication Systems"*, volume **729** of *Lecture Notes in Computer Science***:** 165-202.

12. James, B.A.P (1983), "Variance Reduction Techniques", *Journal of Operational Research Society,* **36**: 525-530.

13. Kleijnen, J.P.C. (1974), *Statistical Techniques in Simulation*, Pt. I, Marcel Dekker, NY.

14. Kleijnen, J.P.C. (1975), "Antithetic Variates, Common Random Numbers and Optimal Computer Time Allocation in Simulation", *Management Science,* **21**: 1176-1185.

15. Kwon, C. and Tew, J.D. (1994), "Strategies for Combining Antithetic Variates and Control Variates in Designed Simulation Experiments", *Management Science,* **40**: 1021-1034.

16. L'Ecuyer, P. (1994), "Efficiency Improvement and Variance Reduction", *Proceedings of the 1994 Winter Simulation Conference*. 122-132.

17. Lavenberg, S.S., Moeller, T.L., and Welch, P.D. (1982), "Statistical Results on Control Variables Applied to the Regenerative Simulation of Queuing Network Simulation", *Operations Research,* **30**: 182-202

18. Law, A.W. (1975), "Efficient Estimators for Simulated Queuing Systems", *Management Science,* **22**: 30-41.

19. Law, A.M. and W.D. Kelton (2000), *Simulation Modeling and Analysis*, McGraw-Hill, NY

20. McKay, M.D., R.J., Beckman, and W.J. Conover. (1979), A comparison of three methods for selecting input random variables in the analysis of output from a computer code. *Technometrics,* **21 (2)**: 239-245.

21. Minh, D.L. (1989), “A Variant of the Conditional Expectation Variance Reduction Technique and Its Application to the Simulation of the GI/G/1 Queues”, *Management Science,* **11**: 1334-1340.

22. Nelson, B.L. (1986), “Variance Reduction Techniques in Computer Simulation”, *Proceedings of the 1986 International Industrial Engineering Conference*, 295-303.

23. Nelson, B.L. (1987), “A Perspective on Variance Reduction in Dynamic Simulation Experiments”, *Communications in Simulations and Statistics-Simulation and Computation*, **B16**: 385-426.

24. Nelson, B.L. (1990), Control Variate Remedies. *Operations Research,* **38 (6)**: 974-992

25. Nelson, B.L., and Schmeiser, B.W. (1983), “Variance Reduction: Basic Transformations”, *Proceedings of the 1983 Winter Simulation Conference*, 255-258.

26. Nelson, B.L., and Schmeiser, B.W. (1986), “Decomposition of Some Well-Known Variance Reduction Techniques”, *Journal of Statistics and Computation in Simulation,* **23**: 183-209.

27. Pegden, C.D., Shannon, R.E., Sadowski, R.P. (1995), *Introduction to Simulation Using SIMAN,* McGraw Hill, NY.

28. Ross, S.M. (1990), *A Course In Simulation,* Macmillan Publishing Company, NY.

29. Ross, S.M. (2001), “Applying Variance Reduction Ideas in Queuing Simulations”, *Probability in the Engineering and Informational Sciences,* **15(4)**: 481-494.

30. Saliby, E. (1997), “Descriptive Sampling: An Improvement Over Latin Hypercube Sampling”, *Proceedings of the 1997 Winter Simulation Conference,* 230-233.

31. Schmeiser, B.W., Taaffe, M.R., and Wang, J. (2001), “Biased Control-Variate Estimation”, *IIE Transactions,* **33**: 219-228.

32. Schruben L.W. and Mangolin, B.H. (1978), “Pseudo-random Number Assignment in Statistically Designed Simulation and Distribution Sampling Experiments”, *Journal of the American Statistical Association,* **73**: 504-525.

33. Sethi, Y.K. (1963), “A Note on Optimum Stratification of Populations for Estimating Population Means”, *Australian J. Statistics,* **5**: 20-33.

34. Stein, M. (1987), Large sample properties of simulations using Latin Hypercube Sampling. *Technnometrics,* **29 (2)**: 143-151.

35. Sullivan, R.S., Hayya, J.C., and Schaul, R. (1982), “Efficiency of the Antithetic Variate Method for Simulating Stochastic Networks”, *Management Science,* **28**: 563-572.

36. Tew, J.D. and Wilson, J.R. (1994), “Estimating Simulation Meta-models Using Combined Correlation-based Variance Reduction Techniques”, *IIE Transactions,* **26**: 2-16.

37. Wilson, J.R. (1984), “Variance Reduction Techniques for Digital Simulation”, *American Journal of Mathematical and Management Science,* **4**: 277-312.

38. Wilson, J.R. and Pritsker, A.A.B. (1984), “Variance Reduction in Queuing Simulation using Generalized Concomitant Variables”, *Journal of Statistics and Computers in Simulation,* **19**: 129-153.

39. Wilson, J.R. and Pritsker, A.A.B. (1994), “Experimental Evaluation of Variance Reduction Techniques for Queuing Simulation using Generalized Concomitant Variables”, *Management Science,* **30**: 1459-1472.

40. Yang, W.N. and Liou, W.W. (1996), “Combining Antithetic Variates and Control Variates in Simulation Experiments”, *ACM Transactions on Modeling and Computer Simulation,* **6**: 243-260.

APPENDICES

APPENDIX A – SIMAN Codes for M/M/1 Queuing System

1. Independent Case

MODEL FILE

```
BEGIN;
                CREATE;
                ASSIGN: ArrRate=5:
                        SerRate=10;
CreateNext      ASSIGN:       UnifSer=UNIF(0,1,3):
                      SerTime= (-1/SerRate)*ln(1-UnifSer):
                      UnifArr=UNIF(0,1,1):
                      IntArrTime= (-1/ArrRate)*ln(1-UnifArr);
               DELAY: IntArrTime;
GoDuplicate     DUPLICATE:1, CreateNext;
               COUNT: JobsIn;
                ASSIGN: NoInSys=NoInSys+1:
                        CustNo=NoInSys:
                        ArrTime=TNOW;
                QUEUE, Buffer;
                SEIZE: Server;
                DELAY: SerTime;
                RELEASE: Server;
                BRANCH, 1:
                        IF, CustNo <= 10000|| CustNo>410000, Warmup:
                        ELSE, NoWarmup;
Warmup          BRANCH, 1:
                        IF, CustNo.EQ.410009, ToFile:
                        ELSE, Destroy;
ToFile          WRITE, 1: TAVG(FlowTime);
                WRITE, 2: TAVG(WaitingTime):NEXT(Destroy);
NoWarmup        TALLY: FlowTime, INT(ArrTime);
                TALLY: WaitingTime, TNOW-ArrTime-SerTime;
Destroy         COUNT: JobsDone:DISPOSE;
END;
```

EXPERIMENTAL FILE

```
BEGIN;
        PROJECT,        MM1;
        ATTRIBUTES:     IntArrTime:ArrTime:SerTime:CustNo:UnifArr:UnifSer;
        VARIABLES:      NoInSys:ArrRate:SerRate;
        QUEUES:         Buffer;
        RESOURCES:      Server;
        TALLIES:        FlowTime: WaitingTime;
        COUNTERS:       JobsDone, 415000: JobsIn, 411000;
        FILES:          Flow, "f1.txt", SEQ, FREE:
                         WaitF, "w1.txt", SEQ, FREE;
;        SEEDS:          1, 23251: 8, 13571;
        REPLICATE,      60,0;
END;
```

2. Antithetic Variates (AV)

MODEL FILE

```
BEGIN;
                CREATE;
                ASSIGN: ArrRate=5:
                        SerRate=10;
CreateNext      ASSIGN:       UnifSer=UNIF(0,1,2):
                      SerTime= (-1/SerRate)*ln(1-UnifSer):
                      UnifArr=UNIF(0,1,1):
                      IntArrTime= (-1/ArrRate)*ln(1-UnifArr);
               DELAY: IntArrTime;
GoDuplicate     DUPLICATE:1, CreateNext;
               COUNT: JobsIn;
                ASSIGN: NoInSys=NoInSys+1:
                        CustNo=NoInSys:
                        ArrTime=TNOW;
                QUEUE, Buffer;
```

```
                SEIZE: Server;
                DELAY: SerTime;
                RELEASE: Server;
                BRANCH, 1:
                        IF, CustNo <= 10000|| CustNo>410000, Warmup:
                        ELSE, NoWarmup;
Warmup          BRANCH, 1:
                        IF, CustNo.EQ.410009, ToFile:
                        ELSE, Destroy;
ToFile          WRITE, 1: TAVG(FlowTime);
                WRITE, 2: TAVG(WaitingTime):NEXT(Destroy);
NoWarmup        TALLY: FlowTime, INT(ArrTime);
                TALLY: WaitingTime, TNOW-ArrTime-SerTime;
Destroy         COUNT: JobsDone:DISPOSE;
END;
```

EXPERIMENTAL FILE

```
BEGIN;
        PROJECT,        MM1;
        ATTRIBUTES:     IntArrTime:ArrTime:SerTime:CustNo:UnifArr:UnifSer;
        VARIABLES:      NoInSys:ArrRate:SerRate;
        QUEUES:         Buffer;
        RESOURCES:      Server;
        TALLIES:        FlowTime: WaitingTime;
        COUNTERS:       JobsDone, 415000: JobsIn, 411000;
        FILES:          Flow, "f1.txt", SEQ, FREE:
                       WaitF, "w1.txt", SEQ, FREE;
;       SEEDS:         1, 23251: 8, 13571;
        REPLICATE,      30,0;
END;
```

3. Latin Hypercube Sampling (LHS) (k=2 and k=3)

a. k=2

MODEL FILE

```
BEGIN;
                CREATE;
                ASSIGN: ArrRate=5:
                        SerRate=10;
CreateNext      ASSIGN: ArrPERM(1)=DISC(1/2,1,1,2,3);
                ASSIGN: ArrPERM(2)=3-ArrPERM(1);
                ASSIGN: SerPERM(1)=DISC(1/2,1,1,2,3);
                ASSIGN: SerPERM(2)=3-SerPERM(1);
               ASSIGN: UnifTime1=(ArrPERM(1)-1+UNIF(0,1,1))/2:
                      UnifTime2=(SerPERM(1)-1+UNIF(0,1,1))/2;
                ASSIGN: IntArrTime=(-1/ArrRate)*ln(1-UnifTime1):
                        SerTime=(-1/SerRate)*ln(1-UnifTime2);
                DELAY: IntArrTime;
GoDuplicate     DUPLICATE:1, CreateNext;
               COUNT: JobsIn;
                ASSIGN: NoInSys=NoInSys+1:
                        CustNo=NoInSys:
                        ArrTime=TNOW;
                QUEUE, Buffer;
                SEIZE: Server;
                DELAY: SerTime;
                RELEASE: Server;
                BRANCH, 1:
                        IF, CustNo <= 10000|| CustNo>410000, Warmup:
                        ELSE, NoWarmup;
Warmup          BRANCH, 1:
                        IF, CustNo.EQ.410009, ToFile:
                        ELSE, Destroy;
ToFile          WRITE, 1: TAVG(FlowTime);
                WRITE, 2: TAVG(WaitingTime):NEXT(Destroy);
NoWarmup        TALLY: FlowTime, INT(ArrTime);
                TALLY: WaitingTime, TNOW-ArrTime-SerTime;
Destroy         COUNT: JobsDone:DISPOSE;
END;
```

EXPERIMENTAL FILE

```
BEGIN;

        PROJECT,        MM1;
        ATTRIBUTES:     ArrTime:SerTime:CustNo:UnifTime1:IntArrTime:
                       UnifTime2:ArrPERM(2):SerPERM(2);
       VARIABLES:     ArrRate:SerRate:NoInSys;
        QUEUES:         Buffer;
        RESOURCES:      Server;
        TALLIES:        FlowTime: WaitingTime;
        COUNTERS:       JobsDone, 415000: JobsIn, 411000;
        FILES:          Flow, "f1.txt", SEQ, FREE:
                        Waiting, "w1.txt", SEQ, FREE;
;      SEEDS:          1, 10357:2, 13597;
        REPLICATE,      30,0;
END;
```

b. k=3

MODEL FILE

```
BEGIN;
                CREATE;
                ASSIGN: ArrRate=9:
                        SerRate=10;
CreateNext      ASSIGN:ArrPERM(1)=DISC(1/3,1,2/3,2,1,3,3);
P21             ASSIGN:ArrPERM(2)=DISC(1/3,1,2/3,2,1,3,3);
                BRANCH,1:
                        If,ArrPERM(1)==ArrPERM(2),P21:
                        Else,GO11;
GO11            ASSIGN:ArrPERM(3)=6-ArrPERM(1)-ArrPERM(2);
                ASSIGN:SerPERM(1)=DISC(1/3,1,2/3,2,1,3,8);
P22             ASSIGN:SerPERM(2)=DISC(1/3,1,2/3,2,1,3,8);
                BRANCH,1:
                        If,SerPERM(1)==SerPERM(2),P22:
                        Else,GO12;
GO12            ASSIGN:SerPERM(3)=6-SerPERM(1)-SerPERM(2);
                ASSIGN: UnifTime1=(ArrPERM(1)-1+UNIF(0,1,1))/3:
                        UnifTime2=(SerPERM(1)-1+UNIF(0,1,1))/3;
                ASSIGN: IntArrTime=(-1/ArrRate)*ln(1-UnifTime1):
                        SerTime=(-1/SerRate)*ln(1-UnifTime2);
                DELAY: IntArrTime;
GoDuplicate     DUPLICATE:1, CreateNext;
                COUNT: JobsIn;
                ASSIGN: NoInSys=NoInSys+1:
                        CustNo=NoInSys:
                        ArrTime=TNOW;
                QUEUE, Buffer;
                SEIZE: Server;
                DELAY: SerTime;
                RELEASE: Server;
                BRANCH, 1:
                        IF, CustNo <= 10000|| CustNo>410000, Warmup:
                        ELSE, NoWarmup;
Warmup          BRANCH, 1:
                        IF, CustNo.EQ.410009, ToFile:
                        ELSE, Destroy;
ToFile          WRITE, 1: TAVG(FlowTime);
                WRITE, 2: TAVG(WaitingTime):NEXT(Destroy);
NoWarmup        TALLY: FlowTime, INT(ArrTime);
                TALLY: WaitingTime, TNOW-ArrTime-SerTime;
Destroy         COUNT: JobsDone:DISPOSE;
END;
```

EXPERIMENTAL FILE

```
BEGIN;

        PROJECT,         MM1;
        ATTRIBUTES:      ArrTime:SerTime:CustNo:UnifTime1:IntArrTime:
                       ArrPERM(3):SerPERM(3):UnifTime2;
       VARIABLES:      NoInSys:ArrRate:SerRate;
        QUEUES:          Buffer;
        RESOURCES:       Server;
        TALLIES:         FlowTime: WaitingTime;
        COUNTERS:        JobsDone, 415000: JobsIn, 411000;
        FILES:           Flow, "f1.txt", SEQ, FREE:
                         Waiting, "w1.txt", SEQ, FREE;
;       SEEDS:         1,10235: 2,25657: 4,26537;
        REPLICATE,       20,0;
END;
```

APPENDIX B – SIMAN Codes for Serial Line Production System

1. Independent Case

MODEL FILE

```
BEGIN;
               CREATE;
CreateNext      ASSIGN: NoInSys=NoInSys+1;
               ASSIGN: CustNo=NoInSys;
               ASSIGN: ServTime(1)=LOGN(1,0.3, 1):
                       ServTime(2)=LOGN(1,0.3, 1):
                       ServTime(3)=LOGN(1,0.3, 1):
                       ServTime(4)=LOGN(1,0.3, 1):
                       ServTime(5)=LOGN(1,0.3, 1);
               ASSIGN: SerTime=ServTime(1)+ServTime(2)+ServTime(3)+
                       ServTime(4)+ServTime(5);
               QUEUE, QueueSet(1);
               SCAN:(NQ(QueueSet(2)).NE.4).AND.(NR(ServerSet(1)).NE.1);
               SEIZE: ServerSet(1);
               DELAY: ServTime(1);
               DUPLICATE:1, CreateNext;
               RELEASE: ServerSet(1);
               COUNT: JobsIn;
               ASSIGN: StationNo=2:ArrTime=TNOW-ServTime(1);
               WHILE: StationNo<5;
                       QUEUE, QueueSet(StationNo),4;
SCAN:(NQ(QueueSet(StationNo+1)).NE.4).AND.(NR(ServerSet(StationNo)).NE.1);
                       SEIZE: ServerSet(StationNo);
                       DELAY: ServTime(StationNo);
                       RELEASE: ServerSet(StationNo);
                       ASSIGN: StationNo=StationNo+1;
               ENDWHILE;
                QUEUE, QueueSet(5),4;
                SEIZE: ServerSet(5);
                DELAY: ServTime(5);
                RELEASE: ServerSet(5);
                BRANCH, 1:
                        IF, CustNo <= 800|| CustNo>40800, Warmup:
                        ELSE, NoWarmup;
Warmup         BRANCH, 1:
                       IF, CustNo.EQ.40809, ToFile:
                       ELSE, Destroy;
ToFile          WRITE, 5: TAVG(ServiceTime1);
               WRITE, 6: TAVG(ServiceTime2);
               WRITE, 7: TAVG(ServiceTime3);
               WRITE, 8: TAVG(ServiceTime4);
               WRITE, 9: TAVG(ServiceTime5);
               WRITE, 3: (2000/(LastTime-Time801));
               WRITE, 1: TAVG(FlowTime);
               WRITE, 2: TAVG(WaitingTime):NEXT(Destroy);
NoWarmup        BRANCH, 1:
                        IF, CustNo.EQ.801, assign801:
                        ELSE, GoTallies;
assign801       ASSIGN: Time801=TNOW-ServTime(1);
GoTallies      TALLY: ServiceTime1, ServTime(1);
               TALLY: ServiceTime2, ServTime(2);
               TALLY: ServiceTime3, ServTime(3);
               TALLY: ServiceTime4, ServTime(4);
               TALLY: ServiceTime5, ServTime(5);
               TALLY: WaitingTime, TNOW-ArrTime-SerTime;
               TALLY: FlowTime, TNOW-ArrTime;
               ASSIGN: LastTime=TNOW;
               ASSIGN: InterDepartTime=TNOW-PrevDepartTime;
               ASSIGN: PrevDepartTime=TNOW;
;              WRITE, 4: InterDepartTime;
Destroy         COUNT: JobsDone:DISPOSE;
END;
```

EXPERIMENTAL FILE

```
BEGIN;
        PROJECT,        MM1;
        ATTRIBUTES:     ArrTime:SerTime:CustNo:StationNo: ServTime(1..5):
                        InterDepartTime;
        VARIABLES:      NoInSys:PrevDepartTime:LastTime:Time801;
        QUEUES:         Buffer1:Buffer2:Buffer3:Buffer4:Buffer5:Buffer6;
        RESOURCES:      Server1:Server2:Server3:Server4:Server5;
        SETS:           QueueSet, Buffer1..Buffer6:
                        ServerSet, Server1..Server5;
        TALLIES:        FlowTime: WaitingTime: ServiceTime1: ServiceTime2:
                        ServiceTime3: ServiceTime4: ServiceTime5;
        COUNTERS:       JobsDone, 42000:JobsIn, 41000;
        FILES:          1, Flow, "f1.txt", SEQ, FREE:
                        2, Wati, "w1.txt", SEQ, FREE:
                        3, Thro, "t1.txt", SEQ, FREE:
                        4, Dept, "tandemDept.txt", SEQ, FREE:
                        5, ser1, "s1-1.txt", SEQ, FREE:
                        6, ser2, "s2-1.txt", SEQ, FREE:
                        7, ser3, "s3-1.txt", SEQ, FREE:
                        8, ser4, "s4-1.txt", SEQ, FREE:
                        9, ser5, "s5-1.txt", SEQ, FREE;
        REPLICATE,      60,0;
END;
```

2. Antithetic Variates (AV)

MODEL FILE

```
BEGIN;
               CREATE;
CreateNext     ASSIGN: NoInSys=NoInSys+1;
               ASSIGN: CustNo=NoInSys;
               ASSIGN: ServTime(1)=LOGN(1,0.3,2):
                       ServTime(2)=LOGN(1,0.3,3):
                       ServTime(3)=LOGN(1,0.3,4):
                       ServTime(4)=LOGN(1,0.3,5):
                       ServTime(5)=LOGN(1,0.3,6);
               ASSIGN: SerTime=ServTime(1)+ServTime(2)+ServTime(3)+
                       ServTime(4)+ServTime(5);
               QUEUE, QueueSet(1);
               SCAN:(NQ(QueueSet(2)).NE.4).AND.(NR(ServerSet(1)).NE.1);
               SEIZE: ServerSet(1);
               DELAY: ServTime(1);
               DUPLICATE:1, CreateNext;
               RELEASE: ServerSet(1);
               COUNT: JobsIn;
               ASSIGN: StationNo=2:ArrTime=TNOW-ServTime(1);
               WHILE: StationNo<5;
                       QUEUE, QueueSet(StationNo),4;
SCAN:(NQ(QueueSet(StationNo+1)).NE.4).AND.(NR(ServerSet(StationNo)).NE.1);
                       SEIZE: ServerSet(StationNo);
                       DELAY: ServTime(StationNo);
                       RELEASE: ServerSet(StationNo);
                       ASSIGN: StationNo=StationNo+1;
               ENDWHILE;
                QUEUE, QueueSet(5),4;
                SEIZE: ServerSet(5);
                DELAY: ServTime(5);
                RELEASE: ServerSet(5);
                BRANCH, 1:
                        IF, CustNo <= 800|| CustNo>40800, Warmup:
                        ELSE, NoWarmup;
Warmup         BRANCH, 1:
                       IF, CustNo.EQ.40809, ToFile:
                       ELSE, Destroy;
ToFile          WRITE, 5: TAVG(ServiceTime1);
               WRITE, 6: TAVG(ServiceTime2);
               WRITE, 7: TAVG(ServiceTime3);
               WRITE, 8: TAVG(ServiceTime4);
               WRITE, 9: TAVG(ServiceTime5);
               WRITE, 3: (2000/(LastTime-Time801));
```

```
                WRITE, 1: TAVG(FlowTime);
                WRITE, 2: TAVG(WaitingTime):NEXT(Destroy);
NoWarmup         BRANCH, 1:
                         IF, CustNo.EQ.801, assign801:
                         ELSE, GoTallies;
assign801        ASSIGN: Time801=TNOW;
GoTallies       TALLY: ServiceTime1, ServTime(1);
                TALLY: ServiceTime2, ServTime(2);
                TALLY: ServiceTime3, ServTime(3);
                TALLY: ServiceTime4, ServTime(4);
                TALLY: ServiceTime5, ServTime(5);
                TALLY: WaitingTime, TNOW-ArrTime-SerTime;
                TALLY: FlowTime, TNOW-ArrTime;
                ASSIGN: LastTime=TNOW;
                ASSIGN: InterDepartTime=TNOW-PrevDepartTime;
                ASSIGN: PrevDepartTime=TNOW;
;               WRITE, 4: InterDepartTime;
Destroy          COUNT: JobsDone:DISPOSE;
END;
```

EXPERIMENTAL FILE

```
BEGIN;
        PROJECT,         MM1;
        ATTRIBUTES:      ArrTime:SerTime:CustNo:StationNo: ServTime(1..5):
                        InterDepartTime;
       VARIABLES:       NoInSys:PrevDepartTime:LastTime:Time801;
        QUEUES:          Buffer1:Buffer2:Buffer3:Buffer4:Buffer5:Buffer6;
        RESOURCES:       Server1:Server2:Server3:Server4:Server5;
       SETS:            QueueSet, Buffer1..Buffer6:
                        ServerSet, Server1..Server5;
        TALLIES:         FlowTime: WaitingTime: ServiceTime1: ServiceTime2:
                        ServiceTime3: ServiceTime4: ServiceTime5;
        COUNTERS:        JobsDone, 42000:JobsIn, 41000;
        FILES:           1, Flow, "f1.txt", SEQ, FREE:
                        2, Wati, "w1.txt", SEQ, FREE:
                         3, Thro, "t1.txt", SEQ, FREE:
                         4, Dept, "tandemAVDept.txt", SEQ, FREE:
                        5, ser1, "s11.txt", SEQ, FREE:
                        6, ser2, "s21.txt", SEQ, FREE:
                        7, ser3, "s31.txt", SEQ, FREE:
                        8, ser4, "s41.txt", SEQ, FREE:
                        9, ser5, "s51.txt", SEQ, FREE;
        SEEDS:          2,,A:3,,A:4,,A:5,,A:6,,A;
       REPLICATE,       60,0;
END;
```

3. Latin Hypercube Sampling (LHS) (k=2 and k=3)

a. k=2

MODEL FILE

```
BEGIN;
                CREATE;
CreateNext      ASSIGN: NMean=ln((LNMean*LNMean)/(sqrt(LNMean*LNMean+LNDev*LNDev))):
                        NDev=SQRT(ln(1+(LNDev/LNMean)*(LNDev/LNMean)));
                ASSIGN: Bound1=EP(0*NDev+NMean);

CreateNext1     ASSIGN: PERM1(1)=DISC(1/2,1,1,2,1);
                 ASSIGN: PERM1(2)=3-PERM1(1);
GoS1             ASSIGN: ServTime(1)=LOGN(LNMean,LNDev,2);
                BRANCH,1:

        IF,((PERM1(pNo)==1)&&(ServTime(1)<Bound1))||((PERM1(pNo)==2)&&(ServTime(1)>Bound1
)),CreateNext2:
                        ELSE, GoS1;

CreateNext2      ASSIGN:PERM2(1)=DISC(1/2,1,1,2,1);
                 ASSIGN:PERM2(2)=3-PERM2(1);
GoS2             ASSIGN: ServTime(2)=LOGN(LNMean,LNDev,2);
                 BRANCH,1:
```

```
        IF,((PERM2(pNo)==1)&&(ServTime(2)<Bound1))||((PERM2(pNo)==2)&&(ServTime(2)>Bound1
)),CreateNext3:
                       ELSE, GoS2;

CreateNext3     ASSIGN:PERM3(1)=DISC(1/2,1,1,2,1);
                ASSIGN:PERM3(2)=3-PERM3(1);
GoS3            ASSIGN: ServTime(3)=LOGN(LNMean,LNDev,2);
                BRANCH,1:

        IF,((PERM3(pNo)==1)&&(ServTime(3)<Bound1))||((PERM3(pNo)==2)&&(ServTime(3)>Bound1
)),CreateNext4:
                        ELSE, GoS3;

CreateNext4     ASSIGN:PERM4(1)=DISC(1/2,1,1,2,1);
                ASSIGN:PERM4(2)=3-PERM4(1);
GoS4            ASSIGN: ServTime(4)=LOGN(LNMean,LNDev,2);
                BRANCH,1:

        IF,((PERM4(pNo)==1)&&(ServTime(4)<Bound1))||((PERM4(pNo)==2)&&(ServTime(4)>Bound1
)),CreateNext5:
                        ELSE, GoS4;

CreateNext5     ASSIGN:PERM5(1)=DISC(1/2,1,1,2,1);
                ASSIGN:PERM5(2)=3-PERM5(1);
GoS5            ASSIGN: ServTime(5)=LOGN(LNMean,LNDev,2);
                BRANCH,1:

        IF,((PERM5(pNo)==1)&&(ServTime(5)<Bound1))||((PERM5(pNo)==2)&&(ServTime(5)>Bound1
)),GoForward1:
                        ELSE, GoS5;

GoForward1     ASSIGN: NoInSys=NoInSys+1;
               ASSIGN: CustNo=NoInSys;
               ASSIGN: SerTime=ServTime(1)+ServTime(2)+ServTime(3)+
                       ServTime(4)+ServTime(5);
;              BRANCH, 1:
;                      IF, CustNo<2900, DuplicateNext:
;                      ELSE, GoQueue;
;DuplicateNext DUPLICATE: 1, CreateNext1;
;GoQueue       COUNT: JobsIn;
               QUEUE, QueueSet(1);
               SCAN:(NQ(QueueSet(2)).LT.4).AND.(NR(ServerSet(1)).NE.1);
               SEIZE: ServerSet(1);
               DELAY: ServTime(1);
               RELEASE: ServerSet(1);
               DUPLICATE:1, CreateNext;
               COUNT:JobsIn;
               ASSIGN: StationNo=2:ArrTime=TNOW-ServTime(1);
               WHILE: StationNo<5;
                       QUEUE, QueueSet(StationNo),4;
SCAN:(NQ(QueueSet(StationNo+1)).LT.4).AND.(NR(ServerSet(StationNo)).NE.1);
                       SEIZE: ServerSet(StationNo);
                       DELAY: ServTime(StationNo);
                       RELEASE: ServerSet(StationNo);
                       ASSIGN: StationNo=StationNo+1;
               ENDWHILE;
                QUEUE, QueueSet(5),4;
                SEIZE: ServerSet(5);
                DELAY: ServTime(5);
                RELEASE: ServerSet(5);
                BRANCH, 1:
                        IF, CustNo <= 800|| CustNo>40800, Warmup:
                        ELSE, NoWarmup;
Warmup         BRANCH, 1:
                       IF, CustNo.EQ.40809, ToFile:
                       ELSE, Destroy;
ToFile          WRITE, 5: TAVG(ServiceTime1);
               WRITE, 6: TAVG(ServiceTime2);
               WRITE, 7: TAVG(ServiceTime3);
               WRITE, 8: TAVG(ServiceTime4);
               WRITE, 9: TAVG(ServiceTime5);
               WRITE, 3: (2000/(LastTime-Time801));
               WRITE, 1: TAVG(FlowTime);
```

```
                 WRITE, 2: TAVG(WaitingTime):NEXT(Destroy);
NoWarmup          BRANCH, 1:
                          IF, CustNo.EQ.801, assign801:
                          ELSE, GoTallies;
assign801         ASSIGN: Time801=TNOW-ServTime(1);
GoTallies        TALLY: ServiceTime1, ServTime(1);
                 TALLY: ServiceTime2, ServTime(2);
                 TALLY: ServiceTime3, ServTime(3);
                 TALLY: ServiceTime4, ServTime(4);
                 TALLY: ServiceTime5, ServTime(5);
                 TALLY: WaitingTime, TNOW-ArrTime-SerTime;
                 TALLY: FlowTime, TNOW-ArrTime;
                 ASSIGN: LastTime=TNOW;
                 ASSIGN: InterDepartTime=TNOW-PrevDepartTime;
                 ASSIGN: PrevDepartTime=TNOW;
;                WRITE, 4: InterDepartTime;
Destroy           COUNT: JobsDone:DISPOSE;
END;
```

EXPERIMENTAL FILE

```
BEGIN;
        PROJECT,         MM1;
        ATTRIBUTES:      ArrTime:SerTime:CustNo:StationNo: ServTime(1..5):
                        InterDepartTime: PERM1(2):PERM2(2):PERM3(2):
                        PERM4(2): PERM5(2);
       VARIABLES:       NoInSys:PrevDepartTime:LastTime:Time801: LNMean,1:
                        LNDev,0.3: NMean: NDev: Bound1: pNo,1;
        QUEUES:          Buffer1:Buffer2:Buffer3:Buffer4:Buffer5:Buffer6;
        RESOURCES:       Server1:Server2:Server3:Server4:Server5;
       SETS:            QueueSet, Buffer1..Buffer6:
                        ServerSet, Server1..Server5;
        TALLIES:         FlowTime: WaitingTime: ServiceTime1: ServiceTime2:
                        ServiceTime3: ServiceTime4: ServiceTime5;
        COUNTERS:        JobsDone, 42000:JobsIn, 41000;
        FILES:           1, Flow, "f1.txt", SEQ, FREE:
                        2, Wati, "w1.txt", SEQ, FREE:
                         3, Thro, "t1.txt", SEQ, FREE:
                         4, Dept, "tandemLHS2Dept.txt", SEQ, FREE:
                        5, ser1, "s11.txt", SEQ, FREE:
                        6, ser2, "s21.txt", SEQ, FREE:
                        7, ser3, "s31.txt", SEQ, FREE:
                        8, ser4, "s41.txt", SEQ, FREE:
                        9, ser5, "s51.txt", SEQ, FREE;
        REPLICATE,      30,0;
END;
```

b. k=3

MODEL FILE

```
BEGIN;
                 CREATE;
CreateNext       ASSIGN: NMean=ln((LNMean*LNMean)/(sqrt(LNMean*LNMean+LNDev*LNDev))):
                        NDev=SQRT(ln(1+(LNDev/LNMean)*(LNDev/LNMean)));
                 ASSIGN: Bound1=EP((-0.4307273)*NDev+NMean):
                        Bound2=EP(( 0.4307273)*NDev+NMean);

CreateNext1      ASSIGN: PERM1(1)=DISC(1/3,1,2/3,2,1,3,1);
P1                ASSIGN: PERM1(2)=DISC(1/3,1,2/3,2,1,3,1);
                  BRANCH,1:
                          IF,PERM1(1)==PERM1(2),P1:
                          Else,GO1;
GO1               ASSIGN:PERM1(3)=6-PERM1(1)-PERM1(2);
GoS1              ASSIGN: ServTime(1)=LOGN(LNMean,LNDev,2);
                 BRANCH,1:

       IF,((PERM1(pNo)==1)&&(ServTime(1)<Bound1))||((PERM1(pNo)==2)&&((ServTime(1)>Bound
1)&&(ServTime(1)<Bound2)))||((PERM1(pNo)==3)&&(ServTime(1)>Bound2)),CreateNext2:
                        ELSE, GoS1;

CreateNext2       ASSIGN: PERM2(1)=DISC(1/3,1,2/3,2,1,3,1);
P2                ASSIGN: PERM2(2)=DISC(1/3,1,2/3,2,1,3,1);
                  BRANCH,1:
```

```
                        IF,PERM2(1)==PERM2(2),P2:
                        Else,GO2;
GO2             ASSIGN:PERM2(3)=6-PERM2(1)-PERM2(2);
GoS2            ASSIGN: ServTime(2)=LOGN(LNMean,LNDev,2);
                BRANCH,1:

       IF,((PERM2(pNo)==1)&&(ServTime(2)<Bound1))||((PERM2(pNo)==2)&&((ServTime(2)>Bound
1)&&(ServTime(2)<Bound2)))||((PERM2(pNo)==3)&&(ServTime(2)>Bound2)),CreateNext3:
                      ELSE, GoS2;

CreateNext3     ASSIGN: PERM3(1)=DISC(1/3,1,2/3,2,1,3,1);
P3              ASSIGN: PERM3(2)=DISC(1/3,1,2/3,2,1,3,1);
                BRANCH,1:
                        IF,PERM3(1)==PERM3(2),P3:
                        Else,GO3;
GO3             ASSIGN: PERM3(3)=6-PERM3(1)-PERM3(2);
GoS3            ASSIGN: ServTime(3)=LOGN(LNMean,LNDev,2);
                BRANCH,1:

       IF,((PERM3(pNo)==1)&&(ServTime(3)<Bound1))||((PERM3(pNo)==2)&&((ServTime(3)>Bound
1)&&(ServTime(3)<Bound2)))||((PERM3(pNo)==3)&&(ServTime(3)>Bound2)),CreateNext4:
                        ELSE, GoS3;

CreateNext4     ASSIGN: PERM4(1)=DISC(1/3,1,2/3,2,1,3,1);
P4              ASSIGN: PERM4(2)=DISC(1/3,1,2/3,2,1,3,1);
                BRANCH,1:
                        IF,PERM4(1)==PERM4(2),P4:
                        Else,GO4;
GO4             ASSIGN: PERM4(3)=6-PERM4(1)-PERM4(2);
GoS4            ASSIGN: ServTime(4)=LOGN(LNMean,LNDev,2);
                BRANCH,1:

       IF,((PERM4(pNo)==1)&&(ServTime(4)<Bound1))||((PERM4(pNo)==2)&&((ServTime(4)>Bound
1)&&(ServTime(4)<Bound2)))||((PERM4(pNo)==3)&&(ServTime(4)>Bound2)),CreateNext5:
                       ELSE, GoS4;

CreateNext5     ASSIGN: PERM5(1)=DISC(1/3,1,2/3,2,1,3,1);
P5              ASSIGN: PERM5(2)=DISC(1/3,1,2/3,2,1,3,1);
                BRANCH,1:
                        IF,PERM5(1)==PERM5(2),P5:
                        Else,GO5;
GO5             ASSIGN: PERM5(3)=6-PERM5(1)-PERM5(2);
GoS5            ASSIGN: ServTime(5)=LOGN(LNMean,LNDev,2);
                BRANCH,1:

       IF,((PERM5(pNo)==1)&&(ServTime(5)<Bound1))||((PERM5(pNo)==2)&&((ServTime(5)>Bound
1)&&(ServTime(5)<Bound2)))||((PERM5(pNo)==3)&&(ServTime(5)>Bound2)),GoForward1:
                        Else, GoS5;

GoForward1     ASSIGN: NoInSys=NoInSys+1;
               ASSIGN: CustNo=NoInSys;
               ASSIGN: SerTime=ServTime(1)+ServTime(2)+ServTime(3)+
                       ServTime(4)+ServTime(5);
               QUEUE, QueueSet(1);
               SCAN:(NQ(QueueSet(2)).LT.4).AND.(NR(ServerSet(1)).NE.1);
               SEIZE: ServerSet(1);
               DELAY: ServTime(1);
               DUPLICATE:1, CreateNext;
               RELEASE: ServerSet(1);
               COUNT:JobsIn;
               ASSIGN: StationNo=2:ArrTime=TNOW-ServTime(1);
               WHILE: StationNo<5;
                       QUEUE, QueueSet(StationNo),4;
SCAN:(NQ(QueueSet(StationNo+1)).LT.4).AND.(NR(ServerSet(StationNo)).NE.1);
                       SEIZE: ServerSet(StationNo);
                       DELAY: ServTime(StationNo);
                       RELEASE: ServerSet(StationNo);
                       ASSIGN: StationNo=StationNo+1;
               ENDWHILE;
                QUEUE, QueueSet(5),4;
                SEIZE: ServerSet(5);
                DELAY: ServTime(5);
                RELEASE: ServerSet(5);
                BRANCH, 1:
```

```
                         IF, CustNo <= 800|| CustNo>40800, Warmup:
                         ELSE, NoWarmup;
Warmup          BRANCH, 1:
                        IF, CustNo.EQ.40809, ToFile:
                        ELSE, Destroy;
ToFile           WRITE, 5: TAVG(ServiceTime1);
                WRITE, 6: TAVG(ServiceTime2);
                WRITE, 7: TAVG(ServiceTime3);
                WRITE, 8: TAVG(ServiceTime4);
                WRITE, 9: TAVG(ServiceTime5);
                WRITE, 3: (2000/(LastTime-Time801));
                WRITE, 1: TAVG(FlowTime);
                WRITE, 2: TAVG(WaitingTime):NEXT(Destroy);
NoWarmup         BRANCH, 1:
                         IF, CustNo.EQ.801, assign801:
                         ELSE, GoTallies;
assign801        ASSIGN: Time801=TNOW-ServTime(1);
GoTallies       TALLY: ServiceTime1, ServTime(1);
                TALLY: ServiceTime2, ServTime(2);
                TALLY: ServiceTime3, ServTime(3);
                TALLY: ServiceTime4, ServTime(4);
                TALLY: ServiceTime5, ServTime(5);
                TALLY: WaitingTime, TNOW-ArrTime-SerTime;
                TALLY: FlowTime, TNOW-ArrTime;
                ASSIGN: LastTime=TNOW;
                ASSIGN: InterDepartTime=TNOW-PrevDepartTime;
                ASSIGN: PrevDepartTime=TNOW;
;               WRITE, 4: InterDepartTime;
Destroy          COUNT: JobsDone:DISPOSE;
END;
```

EXPERIMENTAL FILE

```
BEGIN;
        PROJECT,        MM1;
        ATTRIBUTES:     ArrTime:SerTime:CustNo:StationNo: ServTime(1..5):
                       InterDepartTime: PERM1(3):PERM2(3):PERM3(3):
                       PERM4(3): PERM5(3);
       VARIABLES:      NoInSys:PrevDepartTime:LastTime:Time801: LNMean,1:
                       LNDev,0.3: NMean: NDev: Bound1: Bound2: pNo,1;
        QUEUES:         Buffer1:Buffer2:Buffer3:Buffer4:Buffer5:Buffer6;
        RESOURCES:      Server1:Server2:Server3:Server4:Server5;
       SETS:           QueueSet, Buffer1..Buffer6:
                       ServerSet, Server1..Server5;
        TALLIES:        FlowTime: WaitingTime: ServiceTime1: ServiceTime2:
                       ServiceTime3: ServiceTime4: ServiceTime5;
        COUNTERS:       JobsDone, 42000: JobsIn, 41000;
        FILES:          1, Flow, "f1.txt", SEQ, FREE:
                       2, Wati, "w1.txt", SEQ, FREE:
                        3, Thro, "t1.txt", SEQ, FREE:
                        4, Dept, "tandemLHS3Dept.txt", SEQ, FREE:
                       5, ser1, "s11.txt", SEQ, FREE:
                       6, ser2, "s21.txt", SEQ, FREE:
                       7, ser3, "s31.txt", SEQ, FREE:
                       8, ser4, "s41.txt", SEQ, FREE:
                       9, ser5, "s51.txt", SEQ, FREE;
;       SEEDS:  1,23251:2,13571;
        REPLICATE,     20,0;
END;
```

APPENDIX C – SIMAN Codes for (s,S) Inventory Policy

1. Independent Case

MODEL FILE

```
BEGIN;
       CREATE;
       ASSIGN: InvOnHand=MaxInvLevel:
              InvPosition=MaxInvLevel;
ToLoop WHILE: TNOW<TimeLimit;
              WHILE: InvPosition>ReorderP;
                     DELAY: 1;
                     ASSIGN: Demand=NORM(19.23, 5.658,1);
                     TALLY: DemandT, Demand;
                     ASSIGN: InvPosition=InvPosition-Demand;
                      FINDJ, 1, 25: MIN(DeliveryTime(J));
                      ASSIGN: MinDelivery=J;
                     BRANCH, 1:
                            IF, TNOW>=DeliveryTime(MinDelivery),Arrived:
                            ELSE, NotYet;
Arrived              IF: (InvOnHand+Order(MinDelivery))>Backlogged;
                            COUNT: CycleCount,1;
                            IF: Backlogged.NE.0;
                                   COUNT: BacklogC;
                            ELSE;
                                   COUNT: BacklogC, 0;
                            ENDIF;
                            ASSIGN: InvOnHand=InvOnHand+Order(MinDelivery)-Backlogged;
                            ASSIGN: Backlogged=0;
                            ASSIGN: DeliveryTime(MinDelivery)=1000000;
                     ELSE;
                            COUNT: CycleCount;
                            COUNT: BacklogC,1;
                            ASSIGN: Backlogged=Backlogged-Order(MinDelivery)-
InvOnHand;
                            ASSIGN: InvOnHand=0;
                            ASSIGN: DeliveryTime(MinDelivery)=1000000;
                     ENDIF;
NotYet               IF: InvOnHand>0;
                            IF: InvOnHand>Demand;
                                   ASSIGN: InvOnHand=InvOnHand-Demand;
                                   TALLY: InvOnHandT, InvOnHand;
                                    WRITE, 4: TNOW, InvOnHand;
                            ELSE;
                                   ASSIGN:Backlogged=Backlogged+Demand-InvOnHand;
                                   ASSIGN: InvOnHand=0;
                                   TALLY: InvOnHandT, InvOnHand;
                                    WRITE, 4: TNOW, InvOnHand;
                                   TALLY: BacklogT, Backlogged;
                            ENDIF;
                     ELSE;
                            ASSIGN: Backlogged=Backlogged+Demand;
                            TALLY: BacklogT, Backlogged;
                     ENDIF;
              ENDWHILE;
              ASSIGN: LeadTime=2;
              FINDJ, 1, 25: DeliveryTime(J).EQ.1000000;
              ASSIGN: DeliveryNo=J;
              ASSIGN: DeliveryTime(DeliveryNo)=TNOW+LeadTime;
              ASSIGN: Order(DeliveryNo)=MaxInvLevel-InvPosition;
              ASSIGN: InvPosition=InvPosition+Order(DeliveryNo);
       ENDWHILE:DISPOSE;

       CREATE;
       QUEUE, FakeQueue;
       SCAN: TNOW.EQ.2800;
        WRITE, 1: TAVG(BacklogT);
        WRITE, 2: TAVG(InvOnHandT);
       WRITE, 3: TAVG(DemandT);
       DISPOSE;
END;
```

EXPERIMENTAL FILE

```
BEGIN,,YES;
        PROJECT,        Inv;
        ATTRIBUTES:    Demand:LeadTime:InvPosition:
                       DeliveryNo:MinDelivery;
        VARIABLES:      MaxInvLevel, 150: TimeLimit, 52900: ReorderP,20:
                       DeliveryTime(25), 1000000: Order(25): InvOnHand:
                       Backlogged;
        TALLIES:        DemandT: BacklogT:InvOnHandT;
       QUEUES:         FakeQueue;
       COUNTERS:       CycleCount: BacklogC;
       DSTATS:         InvOnHand, InvOnHandAV:
                       Backlogged, BackloggedAV;
       FILES:          1, BacklogFile, "b1.txt", SEQ, FREE:
                       2, InvOnHandF, "i1.txt", SEQ, FREE:
                       3, DemandF, "d1.txt", SEQ, FREE:
                       4, BackOccF, "backPercent.txt", SEQ, FREE;
        REPLICATE,      1,0,2801, YES, YES, 800;
END;
```

2. Antithetic Variates (AV)

MODEL FILE

```
BEGIN;
       CREATE;
       ASSIGN: InvOnHand=MaxInvLevel:
               InvPosition=MaxInvLevel;
       WHILE: TNOW<TimeLimit;
               WHILE: InvPosition>ReorderP;
                       DELAY: 1;
                       ASSIGN: Demand=NORM(19.23, 5.658,1);
                       TALLY: DemandT, Demand;
                       ASSIGN: InvPosition=InvPosition-Demand;
                        FINDJ, 1, 25: MIN(DeliveryTime(J));
                        ASSIGN: MinDelivery=J;
                       BRANCH, 1:
                               IF, TNOW>=DeliveryTime(MinDelivery),Arrived:
                               ELSE, NotYet;
Arrived                IF: (InvOnHand+Order(MinDelivery))>Backlogged;
                               COUNT: CycleCount,1;
                               IF: Backlogged.NE.0;
                                       COUNT: BacklogC, 1;
                               ELSE;
                                       COUNT: BacklogC, 0;
                               ENDIF;
                               ASSIGN: InvOnHand=InvOnHand+Order(MinDelivery)-Backlogged;
                               ASSIGN: Backlogged=0;
                               ASSIGN: DeliveryTime(MinDelivery)=1000000;
                       ELSE;
                               COUNT: CycleCount;
                               COUNT: BacklogC,1;
                               ASSIGN: Backlogged=Backlogged-Order(MinDelivery)-
InvOnHand;
                               ASSIGN: InvOnHand=0;
                               ASSIGN: DeliveryTime(MinDelivery)=1000000;
                       ENDIF;
NotYet                 IF: InvOnHand>0;
                               IF: InvOnHand>Demand;
                                       ASSIGN: InvOnHand=InvOnHand-Demand;
                                       TALLY: InvOnHandT, InvOnHand;
                               ELSE;
                                       ASSIGN:Backlogged=Backlogged+Demand-InvOnHand;
                                       ASSIGN: InvOnHand=0;
                                       TALLY: InvOnHandT, InvOnHand;
                                       TALLY: BacklogT, Backlogged;
                               ENDIF;
                       ELSE;
                               ASSIGN: Backlogged=Backlogged+Demand;
                               TALLY: BacklogT, Backlogged;
                       ENDIF;
               ENDWHILE;
```

```
                ASSIGN: LeadTime=2;
                FINDJ, 1, 25: DeliveryTime(J).EQ.1000000;
                ASSIGN: DeliveryNo=J;
                ASSIGN: DeliveryTime(DeliveryNo)=TNOW+LeadTime;
                ASSIGN: Order(DeliveryNo)=MaxInvLevel-InvPosition;
                ASSIGN: InvPosition=InvPosition+Order(DeliveryNo);
        ENDWHILE:DISPOSE;

        CREATE;
        QUEUE, FakeQueue;
        SCAN: TNOW.EQ.52800;
        WRITE, 1: TAVG(BacklogT);
        WRITE, 2: TAVG(InvOnHandT);
        WRITE, 3: TAVG(DemandT);
;       WRITE, 4: NC(BacklogC)/NC(CycleCount);
        DISPOSE;
END;
```

EXPERIMENTAL FILE

```
BEGIN;
        PROJECT,        Inv;
        ATTRIBUTES:     Demand:LeadTime:InvPosition:
                        DeliveryNo:MinDelivery;
        VARIABLES:      MaxInvLevel, 150: TimeLimit, 52900: ReorderP,20:
                        DeliveryTime(25), 1000000: Order(25): InvOnHand:
                        Backlogged;
        TALLIES:        DemandT: BacklogT:InvOnHandT;
        QUEUES:         FakeQueue;
        COUNTERS:       CycleCount: BacklogC;
        DSTATS:         InvOnHand, InvOnHandAV:
                        Backlogged, BackloggedAV;
        FILES:          1, Backlog, "b1.txt", SEQ, FREE:
                        2, InvOnHF, "i1.txt", SEQ, FREE:
                        3, DemandF, "d1.txt", SEQ, FREE:
                        4, BackOcF, "backPercentAV.txt", SEQ, FREE;
        SEEDS:         1,,A;
        REPLICATE,      60,0,52801, YES, YES, 800;
END;
```

3. Latin Hypercube Sampling (LHS) (k=2 and k=3)

a. k=2

MODEL FILE

```
BEGIN;
        CREATE;
        ASSIGN: InvOnHand=MaxInvLevel:
                InvPosition=MaxInvLevel:
                Bound1=DemandMean;
        WHILE: TNOW<TimeLimit;
                WHILE: InvPosition>ReorderP;
                        DELAY: 1;
                        ASSIGN: PERM(1)=DISC(1/2,1,1,2,1);
                        ASSIGN: PERM(2)=3-PERM(1);
GoS1                    ASSIGN: Demand=NORM(DemandMean, DemandStdDev,2);
                        BRANCH,1:

        IF,((PERM(pNo)==1)&&(Demand<Bound1))||((PERM(pNo)==2)&&(Demand>Bound1)),GoOn:
                                ELSE, GoS1;
GoOn                    TALLY: DemandT, Demand;
                        ASSIGN: InvPosition=InvPosition-Demand;
                         FINDJ, 1, 25: MIN(DeliveryTime(J));
                         ASSIGN: MinDelivery=J;
                        BRANCH, 1:
                                IF, TNOW>=DeliveryTime(MinDelivery),Arrived:
                                ELSE, NotYet;
Arrived                 IF: (InvOnHand+Order(MinDelivery))>Backlogged;
                                COUNT: CycleCount;
                                IF: Backlogged.NE.0;
                                        COUNT: BacklogC, 1;
                                ELSE;
```

```
                              COUNT: BacklogC, 0;
                        ENDIF;
                        ASSIGN: InvOnHand=InvOnHand+Order(MinDelivery)-Backlogged;
                        ASSIGN: Backlogged=0;
                        ASSIGN: DeliveryTime(MinDelivery)=1000000;
                  ELSE;
                        COUNT: CycleCount;
                        COUNT: BacklogC,1;
                        ASSIGN: Backlogged=Backlogged-Order(MinDelivery)-
InvOnHand;
                        ASSIGN: InvOnHand=0;
                        ASSIGN: DeliveryTime(MinDelivery)=1000000;
                  ENDIF;
NotYet            IF: InvOnHand>0;
                        IF: InvOnHand>Demand;
                              ASSIGN: InvOnHand=InvOnHand-Demand;
                              TALLY: InvOnHandT, InvOnHand;
                        ELSE;
                              ASSIGN:Backlogged=Backlogged+Demand-InvOnHand;
                              ASSIGN: InvOnHand=0;
                              TALLY: InvOnHandT, InvOnHand;
                              TALLY: BacklogT, Backlogged;
                        ENDIF;
                  ELSE;
                        ASSIGN: Backlogged=Backlogged+Demand;
                        TALLY: BacklogT, Backlogged;
                  ENDIF;
            ENDWHILE;
            ASSIGN: LeadTime=2;
            FINDJ, 1, 25: DeliveryTime(J).EQ.1000000;
            ASSIGN: DeliveryNo=J;
            ASSIGN: DeliveryTime(DeliveryNo)=TNOW+LeadTime;
            ASSIGN: Order(DeliveryNo)=MaxInvLevel-InvPosition;
            ASSIGN: InvPosition=InvPosition+Order(DeliveryNo);
      ENDWHILE:DISPOSE;

      CREATE;
      QUEUE, FakeQueue;
      SCAN: TNOW.EQ.52800;
      WRITE, 1: TAVG(BacklogT);
      WRITE, 2: TAVG(InvOnHandT);
      WRITE, 3: TAVG(DemandT);
;      WRITE, 4: NC(BacklogC)/NC(CycleCount);
      DISPOSE;
END;
```

EXPERIMENTAL FILE

```
BEGIN,,YES;
      PROJECT,        Inv;
      ATTRIBUTES:    Demand:LeadTime:InvPosition:
                     DeliveryNo:MinDelivery:PERM(2);
      VARIABLES:      MaxInvLevel, 150: TimeLimit, 52900: ReorderP,20:
                     DeliveryTime(25), 1000000: Order(25): InvOnHand:
                     Backlogged: DemandMean, 19.23: DemandStdDev,5.658:
                     Bound1: pNo,1;
      TALLIES:        DemandT: BacklogT:InvOnHandT;
      QUEUES:         FakeQueue;
      COUNTERS:       CycleCount: BacklogC;
      DSTATS:         InvOnHand, InvOnHandAV:
                      Backlogged, BackloggedAV;
      FILES:          1, Backlog, "b1.txt", SEQ, FREE:
                      2, InvOnHF, "i1.txt", SEQ, FREE:
                      3, DemandF, "d1.txt", SEQ, FREE:
                      4, BackOcF, "backPercentLHS2.txt", SEQ, FREE;
;     SEEDS:         1,23251:2,13571;
      REPLICATE,      30,0,52801, YES, YES, 800;
END;
```

b. k=3

MODEL FILE

```
BEGIN;
       CREATE;
       ASSIGN: InvOnHand=MaxInvLevel:
               InvPosition=MaxInvLevel:
               Bound1=-0.430727*DemandStdDev+DemandMean:
               Bound2= 0.430727*DemandStdDev+DemandMean;
       WHILE: TNOW<TimeLimit;
               WHILE: InvPosition>ReorderP;
                       DELAY: 1;
                       ASSIGN: PERM(1)=DISC(1/3,1,2/3,2,1,3,1);
P1                     ASSIGN: PERM(2)=DISC(1/3,1,2/3,2,1,3,1);
                       BRANCH,1:
                               IF,PERM(1)==PERM(2),P1:
                               Else,GO1;
GO1                    ASSIGN:PERM(3)=6-PERM(1)-PERM(2);
GoS1                   ASSIGN: Demand=NORM(DemandMean, DemandStdDev,2);
                       BRANCH,1:
       IF,((PERM(pNo)==1)&&(Demand<Bound1))||((PERM(pNo)==2)&&((Demand>Bound1)&&(Demand<
Bound2)))||((PERM(pNo)==3)&&(Demand>Bound2)),GoOn:
                               ELSE, GoS1;
GoOn                   TALLY: DemandT, Demand;
                       ASSIGN: InvPosition=InvPosition-Demand;
                        FINDJ, 1, 25: MIN(DeliveryTime(J));
                        ASSIGN: MinDelivery=J;
                       BRANCH, 1:
                               IF, TNOW>=DeliveryTime(MinDelivery),Arrived:
                               ELSE, NotYet;
Arrived                IF: (InvOnHand+Order(MinDelivery))>Backlogged;
                               COUNT: CycleCount;
                               IF: Backlogged.NE.0;
                                       COUNT: BacklogC, 1;
                               ELSE;
                                       COUNT: BacklogC, 0;
                               ENDIF;
                               ASSIGN: InvOnHand=InvOnHand+Order(MinDelivery)-Backlogged;
                               ASSIGN: Backlogged=0;
                               ASSIGN: DeliveryTime(MinDelivery)=1000000;
                       ELSE;
                               COUNT: CycleCount;
                               COUNT: BacklogC,1;
                               ASSIGN: Backlogged=Backlogged-Order(MinDelivery)-
InvOnHand;
                               ASSIGN: InvOnHand=0;
                               ASSIGN: DeliveryTime(MinDelivery)=1000000;
                       ENDIF;
NotYet                 IF: InvOnHand>0;
                               IF: InvOnHand>Demand;
                                       ASSIGN: InvOnHand=InvOnHand-Demand;
                                       TALLY: InvOnHandT, InvOnHand;
                               ELSE;
                                       ASSIGN:Backlogged=Backlogged+Demand-InvOnHand;
                                       ASSIGN: InvOnHand=0;
                                       TALLY: InvOnHandT, InvOnHand;
                                       TALLY: BacklogT, Backlogged;
                               ENDIF;
                       ELSE;
                               ASSIGN: Backlogged=Backlogged+Demand;
                               TALLY: BacklogT, Backlogged;
                       ENDIF;
               ENDWHILE;
               ASSIGN: LeadTime=2;
               FINDJ, 1, 25: DeliveryTime(J).EQ.1000000;
               ASSIGN: DeliveryNo=J;
               ASSIGN: DeliveryTime(DeliveryNo)=TNOW+LeadTime;
               ASSIGN: Order(DeliveryNo)=MaxInvLevel-InvPosition;
               ASSIGN: InvPosition=InvPosition+Order(DeliveryNo);
       ENDWHILE:DISPOSE;

        CREATE;
        QUEUE, FakeQueue;
        SCAN: TNOW.EQ.52800;
```

```
        WRITE, 1: TAVG(BacklogT);
        WRITE, 2: TAVG(InvOnHandT);
        WRITE, 3: TAVG(DemandT);
;        WRITE, 4: NC(BacklogC)/NC(CycleCount);
        DISPOSE;
END;
```

EXPERIMENTAL FILE

```
BEGIN;
        PROJECT,        Inv;
        ATTRIBUTES:    Demand:LeadTime:InvPosition:
                       DeliveryNo:MinDelivery:PERM(3):Bound1:Bound2;
        VARIABLES:      MaxInvLevel, 150: TimeLimit, 52900: ReorderP,20:
                       DeliveryTime(25), 1000000: Order(25): InvOnHand:
                       Backlogged:DemandMean, 19.23: DemandStdDev, 5.658:
                       pNo,1;
        TALLIES:        DemandT: BacklogT:InvOnHandT;
        QUEUES:         FakeQueue;
        COUNTERS:       CycleCount: BacklogC;
        DSTATS:         InvOnHand, InvOnHandAV:
                        Backlogged, BackloggedAV;
        FILES:          1, Backlog, "b1.txt", SEQ, FREE:
                        2, InvOnHF, "i1.txt", SEQ, FREE:
                        3, DemandF, "d1.txt", SEQ, FREE:
                        4, BackOcF, "backPercentLHS3.txt", SEQ, FREE;
;       SEEDS:         1,23251:2,13571;
        REPLICATE,      20,0,52801, YES, YES, 800;
END;
```