IMPORTANCE SAMPLING FOR QUEUEING SIMULATIONS

by

Semih Yön

B.S., Industrial Engineering, Yıldız Technical University, 2004

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University
2007

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge the enthusiastic supervision of Assoc. Prof. Wolfgang Hörmann. His guidance, insightful criticisms and patient encouragement contributed very much for the realization of this thesis.

I would also like to acknowledge the support of TÜBİTAK, the unique research foundation in Turkey, with the program of BAYG 2210.

I want to acknowledge the LaTeX helps of Onur Özgün and Engin Durmaz during the writing period of this thesis.

And finally I want to specially thank to my favorite supporters at every time: my dear mum Selmin Yön, dear dad Hasan Yön and my whole family.

ABSTRACT

IMPORTANCE SAMPLING FOR QUEUEING SIMULATIONS

In a simulation it is desired to control variability and decrease the variance of experiments. So we can be aware of the accuracy of constructed models and consequently can supply reliable results. Importance Sampling is one of the variance reduction techniques commonly used in Monte Carlo Methods. There are two types: Independent Importance Sampling (IIS) where iid random variables are used to calculate some expectations whereas the other one is Dependent Importance Sampling (DIS) where dependent random variables are used. This thesis includes a research to find the Importance Sampling density which gives the lowest variance. We illustrate the Importance Sampling method on an M/M/1 queueing problem with a finite upstreambuffer and solve it with an efficient C coded simulation program. We first execute naïve simulation, afterwards we carry out Importance Sampling and reach a meaningful decrease in the estimated variance when calculating the probability that the queue length exceeds the buffer size. Thus, one can calculate any expectation with good confidence intervals that cannot be calculated analytically. Numerical results indicate that heavy tailed Importance Sampling distributions provide substantial variance reduction.

ÖZET

KUYRUK AĞI BENZETİM UYGULAMALARI İÇİN ÖNEMİNE GÖRE ÖRNEKLEME YÖNTEMİ

Benzetim yönteminde değişkenliği kontrol etmek ve deneylerin varyansını azaltmak arzulanır. Böylece kurulan modellerin doğruluğundan haberdar olabilir ve neticede güvenilir sonuçlar elde edebiliriz. Önemine göre Örnekleme Monte Carlo metotlarında sıkça kullanılan varyans azaltma tekniklerinden bir tanesidir. Önemine göre Örnekleme Metodu iki çeşit olarak görülebilir, birincisi olan Bağımsız Önemine göre Örneklemede bağımsız ve aynı dağılımdan alınmış rassal değişkenler kullanılarak birtakım beklenen değerler hesaplanırken ikincisi olan Koşullu Önemine göre Örneklemede birbirine bağımlı rassal değişkenler kullanılır. Bu tez en düşük varyansı veren en iyi Önemine göre Örnekleme dağılımını elde etmek üzere bir araştırmayı içermektedir. Önemine göre Örnekleme metodunu tek servisli kısıtlı bekleme kapasiteli bir Markovian kuyruk problemi üzerinde gösterdik ve bunu C dilinde etkin bir benzetim programı ile çözdük. Önce basit Monte Carlo benzetim yöntemini daha sonra da Önemine göre Örnekleme Metodunu uyguladık ve kuyruk uzunluğunun tampon kapasitesini geçtiği durumlar için anlamlı düzeyde varyans azalması sağladık. Bu yöntemle analitik olarak hesaplanamayan bir takım beklenen değerler hesaplanabilir. Sayısal sonuçlar dolgun kuyruklara sahip Önemine göre Örnekleme dağılımlarının daha iyi sonuçlar verdiğini göstermiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

φ	Probability for selecting the sampling region
λ	Rate of the Exponential interarrival times distribution
μ	Mean of the Exponential service times distribution
ν	Rate of the Exponential service times distribution
θ	Expectation value for Monte Carlo integral
ρ	Traffic intensity
ψ	Limit for the unimportant region
AvrgW	Average Weight
HL	Half Length
iid	Independent and identically distributed
IS	Importance Sampling
LB	Lower Bound
MaxW	Maximum Weight
MinW	Minimum Weight
MSE	Mean Squared Error
n	Number of Replications of Experiments
pdf	Probability Density Function
UB	Upper Bound
2A0B	Two Arrivals and Buffer Size of Zero Case
3A1B	Three Arrivals and Buffer Size of One Case
11A5B	Eleven Arrivals and Buffer Size of Five Case
101A20B	Hundredone Arrivals and Buffer Size of Twenty Case
101A22B	Hundredone Arrivals and Buffer Size of Twentytwo Case

1. INTRODUCTION

Simulations driven by random inputs will produce random outputs. If we can somehow reduce the variance of an output random variable of interest without disturbing its expectation, we should have greater precision that means shorter confidence intervals. The classical Monte Carlo method (Naïve Simulation) is a random number based approach to estimate quantities that are hard to compute exactly. Monte Carlo methods can be viewed as numerically approximating integrals and have many application areas such as global optimization, nuclear shielding, queueing networks, highly reliable systems and computational finance. Many problems in these areas can be formulated as integrals over a single model distribution or highly multi-modal distributions in the result of expectations. The accuracy of this estimation strongly depends on the quality of sampling which can be improved in two ways:

- increasing the cardinality of sampling or
- introducing some kind of selection rules that make sample more representative.

To make Monte Carlo calculations faster and improve the accuracy, a biased sampling with weight coefficients, so called Importance Sampling (IS) is used. The contribution of IS is to introduce definite selection rules to generate the most likely samples or configurations and hence to obtain more accurate values of statistical averages.

The IS method introduces a different density which we call *importance sampling density* throughout this thesis. The IS density can be selected close to the original density or sometimes freely. The most important feature of the IS density towards the original density is that it should sample more from the important region of the integral. The IS density makes rare events less rare and assigns relatively smaller weights to get an unbiased estimator. Thus, IS performs well for especially rare event simulations.

There is a large literature on IS applications. For example: Hesterberg [1], proposes defensive mixture IS densities for oil inventory reliability at a large utility. And

Hesterberg calls the original density *target distibution* and the IS density *design distribution.* Basamboo *et al.* [2], proove that there does not exist an asymptotically optimal state independent IS density for estimating the probability of a random walk with heavy-tailed increments. And they introduce a state-dependent IS estimator. Yuan and Druzdzel [3], use IS for Bayesian Networks and call the original density *posterior distribution* and the IS density *importance function* to indicate that the IS density should sample more from the important region. Campioni *et al.* [4], use IS technique with exponential and uniform biasing to optimize the unreliability assessment of an exponential component. Ross and Wang [5], apply IS on product-form queueing networks and they propose the software MonteQueue 2.0 for calculating the key performance measures of queueing networks. Kuruganti and Strickland [6], focuse on computing the optimal IS measures of the stochastic processes in a tandem queueing system. Suzuki and Nakagawa [7], use IS for detecting cell loss probabilities lower than 10^{-12} in Asynchronous Transfer Mode (ATM) switch including input and output buffers with back pressure control. And they call the IS density *simulated distribution.* Xiao *et al.* [8], use IS for dependability estimation of the reliability and transient availability of a non-Markov consecutive-k-out-of-n system. Hurtado [9], proposes an algorithm for estimating the failure probability of a structural reliability system combining IS and pattern recognition techniques. Chen [10], shows that Rejection Sampling can be viewed as a special IS algorithm. Bekaert *et al.* [11], use IS for stochastic Jacobi Radiosity system solution. They call the original distribution *target pdf* and the IS distribution *source pdf.* Evans and Swartz [12], give a detailed chapter on Independent Importance Sampling and the theory behind it.

The IS method is commonly used as a variance reduction technique. In order to reduce variance a product form correction factor is computed based on sample weights accumulated during sampling. With proper weights the correction factor compensates for statistical fluctuations and leads to a lower variance. Very commonly used variance reduction techniques are IS, stratification, common random numbers, antithetic variates and control variates. The first two methods reflect the idea of using weighted sampling based on a priori qualitative or quantitative information in an attempt to reduce variance whereas the others concentrate on introducing correlation to reduce variance. IS along other variance reduction techniques is discussed by Bratley *et al.*[13].

IS is valuable when it is impossible or too expensive to generate random samples from the original density of Naïve Monte Carlo experiment. Then we cannot use naïve simulation. Hence, IS can provide solutions to the problems besides leading to a variance reduction. We can calculate variance estimates (MSE) for different parameters of the IS density and can minimize the MSE in a pilot study.

The aim of this thesis is to introduce IS method with the theory behind it and to find the optimal IS density for simple queueing systems. For high dimensions IS is difficult to implement, so we applied the IS method on an M/M/1 queueing system with finite upstream buffer for best possible variance reductions. The organisation of the thesis is as follows: Chapter 2 gives the IS method definition with detailed selection rules and some examples of IS densities. Then in Chapter 3 we explain the implementaion of Independent Importance Sampling for a simple M/M/1 queueing system. Chapter 4 introduces multivariate Dependent Importance Sampling with implementation for the M/M/1 queueing system. Furthermore we propose a new sampling approach called *Cut Method* in Chapter 4. We cut the sampling region and get the important part and then just sample from there. Chapter 5 explains the implementation of IS. We implement IS with C and ARENA. And finally; Conclusions, Appendix and References are set at the end of the thesis, respectively.

2. IMPORTANCE SAMPLING METHOD

2.1. Method Definition

Importance Sampling (IS) is used for numerically approximating integrals and also seen as a variance reduction technique in Monte Carlo applications. The general Monte Carlo integral can be shown as:

$$\theta = E_f\left[q\left(\mathbf{X}\right)\right] = \int_{R^d} q\left(\mathbf{X}\right) f\left(\mathbf{x}\right) d\mathbf{x} \tag{2.1}$$

where $\mathbf{X} = (x_1, .., x_d)$ denotes a vector of iid random variables in d dimensional sampling space (R^d), and $\mathbf{X}$ has a joint density function (or joint mass function in the discrete case) of $f\left(\mathbf{x}\right) = f\left(x_1, .., x_d\right)$. The function $q\left(\mathbf{X}\right)$ in equation 2.1 is an arbitrary real valued function in the sampled region.

The idea behind IS is that certain values of the input random variables in a simulation have more impact on the parameter being estimated than others. If these important values are emphasized by sampling more frequently, then the estimator variance can be reduced. Hence, the basic methodology in IS is to choose a distribution which encourages the important values. This use of a different distribution would result in a biased estimator. However, the simulation outputs are weighted to correct for the use of the different distribution, and this ensures that the new IS estimator is unbiased. We can describe the application of IS by using

$$\theta = E_f\left[q\left(\mathbf{X}\right)\right] = E_g\left[q\left(\mathbf{X}\right)\frac{f\left(\mathbf{x}\right)}{g(\mathbf{x})}\right] = \int_{R^d} q\left(\mathbf{X}\right)\frac{f\left(\mathbf{x}\right)}{g(\mathbf{x})} g\left(\mathbf{x}\right) d\mathbf{x} \tag{2.2}$$

where $g\left(\mathbf{x}\right)$ is the IS density such that $g\left(\mathbf{x}\right) \neq 0$ whenever $f\left(\mathbf{x}\right) \neq 0$. The notations E_f and E_g indicate that the expectation is taken with respect to a random vector $\mathbf{X}$ with joint densities $f\left(\mathbf{x}\right)$ and $g\left(\mathbf{x}\right)$, respectively. A Monte Carlo algorithm to evaluate an integral is to sample a sequence of x_i forming random vectors of $\mathbf{X}$ with a density like

$$f(\mathbf{x}) = \prod_{i=1}^{d} f(x_i) \tag{2.3}$$

Then we have the estimator $\hat{\theta}$ for the Monte Carlo integration with n samples:

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} q(\mathbf{X_i})\, w(\mathbf{X_i}) \tag{2.4}$$

where $w(\mathbf{x}) = \frac{f(\mathbf{x})}{g(\mathbf{x})}$ is the correction factor also called *weight function.* As Importance Sampling is a variance reduction technique, the variance formula for the integral θ is of high importance for reliability of IS. The theoretical variance formula whose derivation can be found in Appendix A is as follows:

$$var(\theta) = \int_{R^d} \left(q(\mathbf{X})^2\right) \frac{f(\mathbf{x})}{g(\mathbf{x})} f(\mathbf{x})\, d\mathbf{x} - \boldsymbol{\theta^2} \tag{2.5}$$

For selection of a sensible density $g(x)$, the estimator $\hat{\theta}$ in equation (2.4) should have some important properties, Fishman [14].

- The first one is based on the *Law of Large Numbers.* If the sample size $n \to \infty$, the estimator $\hat{\theta}$ converges almost surely to expectation θ.

$$P\left\{lim_{n\to\infty}\, \hat{\theta} = \theta\right\} = 1. \tag{2.6}$$

- Secondly, if the sampling process is independent, we have iid data and also have

$$Var(\hat{\theta}) = \frac{Var(\theta)}{n} \tag{2.7}$$

- Third, for any fixed sample size n there is a pdf that describes the values of $\hat{\theta}$ and as $n \to \infty$ this pdf becomes the standard normal distribution. The *Central Limit Theorem* shows that the random variable in brackets below is distributed standard normal.

$$\left(\frac{\hat{\theta} - \theta}{\sqrt{\frac{Var\theta}{n}}} \right) \xrightarrow{d} N(0,1),\ as\ n \rightarrow \infty$$

After setting the theory behind Monte Carlo Concepts and IS applications we can discuss both the general rules for selecting IS densities and the properties of the weight function in detail in section (2.2).

2.2. General Rules for Selecting the Importance Sampling Density

The interpretation of IS shown in equation (2.2) requires an optimal IS density for minimizing the variance. Kalos [15], showed the following result using equation (2.5) and a Lagrange multiplier.

Result:

The optimal IS density can be written as

$$g^{*}(\mathbf{x}) = \frac{|q(\mathbf{X})| f(\mathbf{x})}{\theta}. \tag{2.8}$$

This IS density leads to a zero variance estimator. However, it requires the knowledge of θ, the result we are looking for. So, in practise the optimal IS density shown in equation (2.8) is not useful. Then the selection of an IS density is a process including decisions. The first important rule is that the IS density must be nonzero whenever the original density is nonzero, otherwise both the weight function and the variance may be unbounded.

A first choice of the IS density should be roughly proportional to $f(\mathbf{x})$ but in the way of sampling more from region where $q(\mathbf{X})$ is large. Otherwise the variance of IS can be larger than variance of the Naïve Monte Carlo. In an attempt to minimize the variance, it is easy to see that the IS density should be as large as possible in the important region. And conversely, in the unimportant region IS density should be allowed to be smaller. Because we have the constraint of

$$\int_{R^d} g(\mathbf{x})\, d\mathbf{x} = \mathbf{1} \tag{2.9}$$

in the d-dimensional sampling space.

When the IS method is used for estimating the expectation of a rare event, the IS density should in particular sample more in the region where $q(\mathbf{X})$ is large and $f(x)$ is small. In other words the IS density should change the sampling frequency of rare events in favour of the estimator $\hat{\theta}$ for a fixed sample size. It should increase the frequency of the rare event. For instance, if we investigate the expectation of a buffer overflow in a queueing system, one way to increase the frequency of buffer overflows is to select an IS density that increases the service times. In Figure 2.1 we show two possible alternative IS densities for Exponential($\lambda = 1$) original density. Both, Exponential($\lambda = 0.5$) and shifted Pareto($\alpha = 1; \beta = 1$) IS densities have higher tails than the original density. When the Pareto distribution is shifted as much as its scale parameter β, it is similar to the Exponential distribution. We show the form of the shifted Pareto IS density in section 2.3.2.

Another way to increase the frequency of buffer overflows is to select an IS density that decreases the interarrival times in a queueing system. In Figure 2.2 we show two possible alternative IS densities for Exponential($\lambda = 0.7$) original density. The important region is then the area below the curves close to the origin. For the important region, IS densities sample more and we have the likelihood ratio $\frac{f(x)}{g(x)}$ relatively smaller. Often IS is used for calculating the expectations of rare events occuring in the tails of distributions. So we have heavy tailed IS densities similar to our first example. But now we consider the left tails which are x-values close to zero for these distributions.

An IS density which leads to a good estimator for function $q(\mathbf{X})$ may also be suitable for different $q(\mathbf{X})$ functions at the same time. For instance, an IS density which is optimal for buffer exceedance probability in an M/M/1 queueing system, may

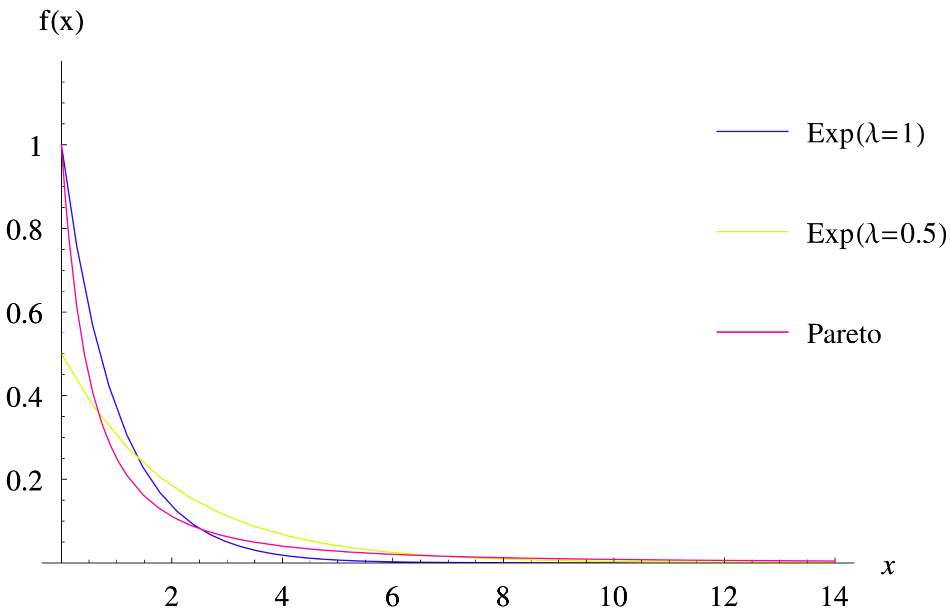


Figure 2.1. Importance Sampling densities useful for service times - Higher tailed distributions

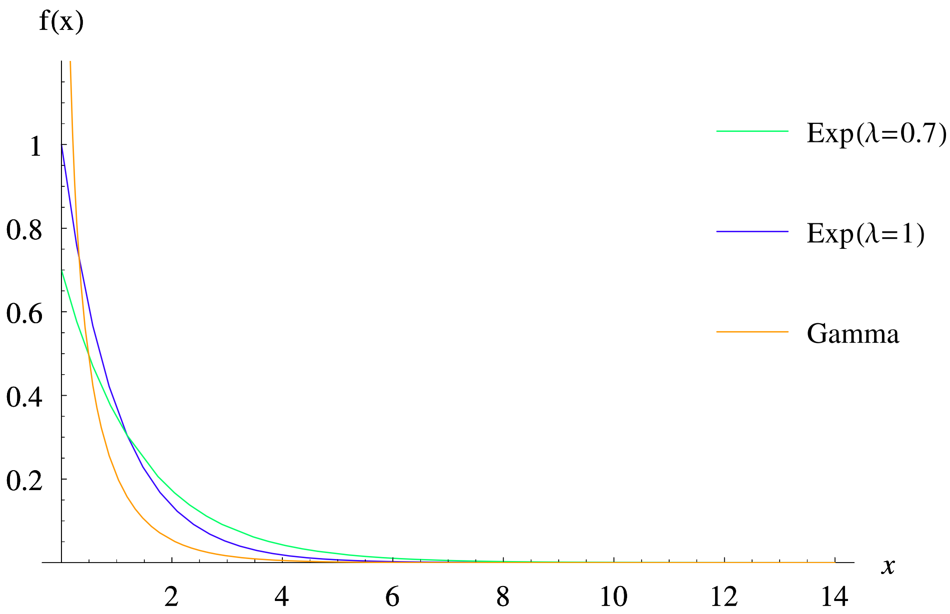


Figure 2.2. Importance Sampling densities useful for interarrival times

be also useful for the expectation of the maximum waiting time in the queue. Or the opposite may be true for some applications that if the sampling region can be seperated in two or more parts, different IS densities might be optimal for each region. The general shape of the sampling region should be analyzed while selecting the IS density.

Another highly important point while selecting an IS density is the weight function. The weight function plays a key role in the accuracy of IS as it is one component of both the variance formula and the summation for estimator $\hat{\theta}$. The IS density should provide relatively small weights in important regions because it is already increasing the frequency of rare events in important regions. Although large weights would be eliminated for the unimportant region of the sample space with very small values of $q(\mathbf{X})$, the weight function should not be extremely large. Otherwise, the random number based approach of Monte Carlo Simulations may disturb the estimator via large values of weight function. That is why the weight function should be bounded above in order not to accidently have very large values of estimators or infinitely large variances. To prevent this, Hesterberg [1] proposes defensive mixture distributions that we discuss in section (2.3.4). The IS density with smaller maximum weight value is prefered among two similar IS densities. Although the likelihood ratio $\frac{f(\mathbf{x})}{g(\mathbf{x})}$ will be smaller than 1 in large regions, the average weight is obviously 1 as $f(\mathbf{x})$ and $g(\mathbf{x})$ are both densities.

2.3. Importance Sampling Densities Useful for Exponential Input Distributions

The Exponential distribution is of greatest importance. When Exponential input distributions are used for IS, it is so easy to design and implement the model. Nevertheless, by increasing the frequency of important events, it is possible to accurately estimate the frequency or expectation of some rare events with better confidence intervals or preferably with fewer Monte Carlo replications. For instance, the upstream buffer overflow probability increases with either the increase of arrival rate or decrease of service rate in an M/M/1 queueing system.

We use Exponential IS densities with different rates, shifted Pareto(α, β) IS densities with $\alpha > 0, \beta > 0$, Gamma(α, β) IS densities with $\alpha = 0.5$, $\alpha = 0.9$ and any $\beta > 0$ and defensive mixture distributions for higher tails.

2.3.1. Exponential Importance Sampling Density "Changing the Rate"

As the application area of IS in Monte Carlo concepts is wide and some are vital in their topics, the reliability of the results with IS becomes of high importance. Although some computational statistical tools allow to define satisfactory confidence intervals and some quantiles of interest, the nature of the methodology with random number generation (or generation of vectors consisting of random numbers) makes it obligatory to show the stability of IS. We selected two easy cases for the interpretation of the theoretical use of IS as first examples.

Example 1:

As first very simple illustration of Importance Sampling we calculate the expectation of an exponential random variable with mean $\mu = 1$. We used exponential IS densities with different rates of λ. Then in equation (2.2) we have $q(X) = x$, and equation (2.2)

$$\theta = \int_0^\infty x \frac{e^{-x}}{\lambda e^{-\lambda x}} \left(\lambda e^{-\lambda x}\right) dx \tag{2.10}$$

The results of Naïve Monte Carlo experiment and IS are displayed in Table 2.1. As it is known that the result is one, we have the correct estimator for all parameters of the IS density and write:

$$\hat{\theta}_\lambda \cong 1.0, \forall\lambda \tag{2.11}$$

In our experiment we observed that the lowest variance (MSE) is achieved with $\lambda = 0.50$ as $var_{(\lambda=0.5)}(\hat{\theta}) = 0.18484$ where naïve variance is $var_{naive} = 0.99823$. When we look at the amount of variance reduction, we have a reduction factor

$$\frac{var_{Naive}}{var_{IS}} = \frac{0.99823}{0.18484} = 5.4 \tag{2.12}$$

Table 2.1. Results for the expectation of an exponential random variable

λ	$\hat{\theta}$	MSE	HL	LB	UB	AW	MinW	MaxW
Naïve Results								
1.00	1.00040	0.99823	0.00196	0.99845	1.00236	1.00000	1.00000	1.00000
IS Results								
0.01	0.99857	24.32590	0.00967	0.98890	1.00824	1.00774	0.00000	99.99469
0.10	0.99833	1.91497	0.00271	0.99562	1.00104	0.99425	0.00000	9.99995
0.20	1.00130	0.71461	0.00166	0.99965	1.00296	1.00086	0.00000	4.99989
0.30	1.00018	0.35659	0.00117	0.99901	1.00135	1.00209	0.00000	3.33333
0.40	0.99965	0.22140	0.00092	0.99873	1.00057	0.99795	0.00000	2.49999
0.41	1.00006	0.21409	0.00091	0.99915	1.00097	1.00023	0.00000	2.43903
0.42	0.99962	0.20651	0.00089	0.99873	1.00051	1.00025	0.00000	2.38095
0.43	1.00000	0.20118	0.00088	0.99912	1.00088	1.00073	0.00000	2.32558
0.44	0.99989	0.19632	0.00087	0.99902	1.00076	1.00021	0.00000	2.27272
0.45	1.00084	0.19188	0.00086	0.99999	1.00170	1.00040	0.00000	2.22222
0.46	1.00028	0.18948	0.00085	0.99943	1.00113	0.99990	0.00000	2.17391
0.47	1.00025	0.18744	0.00085	0.99940	1.00110	0.99990	0.00000	2.12766
0.48	0.99959	0.18594	0.00085	0.99875	1.00044	1.00018	0.00000	2.08333
0.49	1.00074	0.18506	0.00084	0.99990	1.00158	0.99946	0.00000	2.04082
0.50	1.00081	**0.18484**	0.00084	0.99997	1.00166	0.99904	0.00000	2.00000
0.51	1.00004	0.18554	0.00084	0.99920	1.00089	1.00042	0.00000	1.96078
0.52	1.00039	0.18596	0.00085	0.99955	1.00124	0.99866	0.00000	1.92308
0.53	1.00009	0.18751	0.00085	0.99924	1.00094	0.99992	0.00000	1.88679
0.54	0.99975	0.18948	0.00085	0.99889	1.00060	0.99961	0.00000	1.85185
0.55	1.00024	0.19148	0.00086	0.99938	1.00110	1.00056	0.00001	1.81818
0.56	1.00078	0.19479	0.00087	0.99991	1.00164	0.99951	0.00002	1.78572
0.57	1.00009	0.19913	0.00087	0.99922	1.00096	0.99966	0.00002	1.75439
0.58	0.99976	0.20401	0.00089	0.99887	1.00064	0.99939	0.00006	1.72414
0.59	0.99953	0.20983	0.00090	0.99863	1.00042	1.00030	0.00011	1.69492
0.60	0.99894	0.21573	0.00091	0.99803	0.99985	1.00060	0.00021	1.66667
0.70	1.00057	0.29987	0.00107	0.99949	1.00164	0.99996	0.00264	1.42857
0.80	0.99919	0.44556	0.00131	0.99788	1.00050	1.00053	0.02570	1.25000
0.90	0.99845	0.66896	0.00160	0.99685	1.00005	1.00036	0.20925	1.11111
1.00	1.00022	0.99463	0.00195	0.99827	1.00218	1.00000	0.99999	1.00000

The graph of the variance with respect to λ is *u-shaped* as shown in Figure 2.3.

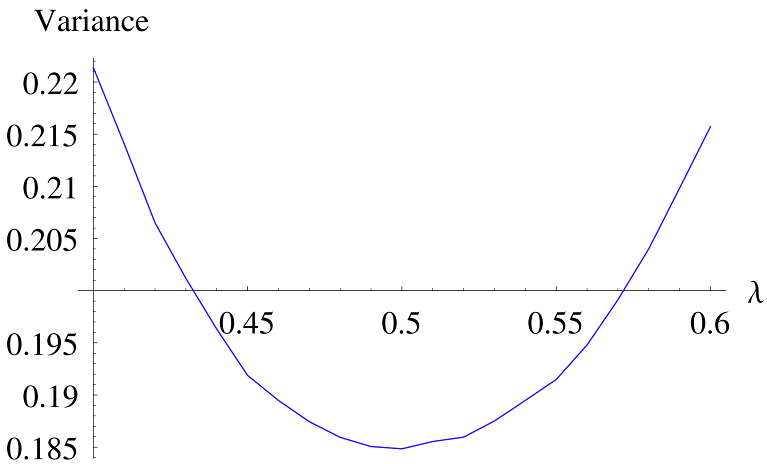


Figure 2.3. Variance graph for example 1

We can even prove that the optimal rate for exponential IS density is $\lambda = 0.50$. We have a general variance formula for our example as in the following and can calculate the integral easily by parts with L'Hospital rule:

$$\begin{aligned} var(x) &= \int_0^\infty x^2 \frac{e^{-2x}}{\lambda e^{-\lambda x}} dx - \theta^2 \\ &= \frac{1}{\lambda} \frac{-2}{(\lambda - 2)^3} - \theta^2 \end{aligned} \tag{2.13}$$

To find the optimal parameter (λ^*) that minimizes the equation (2.13) the first derivative with respect to λ is calculated.

$$\frac{\partial var_\lambda(x)}{\partial \lambda} = \frac{4(2\lambda - 1)}{\lambda^2 (\lambda - 2)^4} = 0 \tag{2.14}$$

Then we have optimal $\lambda^* = 0.5$ which ensures our IS results. The results shown in Table 2.1 are as expected. We have all average weights so close to 1. Moreover, IS density with rate $\lambda = 1$ gives 1 for both maximum weight and minimum weight. Visual profit of IS appears in the confidence intervals. The naïve 95 percent confidence interval is

$$0.99845 \leq \hat{\theta} \leq 1.00236 \tag{2.15}$$

whereas 95 percent confidence interval with IS is

$$0.99997 \leq \hat{\theta} \leq 1.00166 \tag{2.16}$$

When we look at Table 2.1 we see that the maximum weights decrease from $\lambda = 0.1$ to $\lambda = 1$ and all are not very large.

Example 2:

As it is mentioned in section 2.1 Monte Carlo Simulations are more important for calculation of higher dimensional integrals. We therfore demonstrate the use of IS with 2 dimensions. If we calculate the expectation of the maximum of two Exponential random variables, we would get the two dimensional integral. For the simplicity we have a random vector $\mathbf{x}$: $\{x_1, x_2\}$ for two iid standard Exponential varietes. Then the integral is

$$\theta = \int_0^\infty \int_0^{x_1} x_1 \prod_{i=1}^{2} \frac{e^{-x_i}}{\lambda e^{-\lambda x_i}} \prod_{i=1}^{2} \lambda e^{-\lambda x_i} dx_2 dx_1 \tag{2.17}$$

with the assumption of $x_1 > x_2$. And the theoretical variance can be calculated using Mathematica software:

$$\begin{aligned} var(x) &= \int_0^\infty \int_0^{x_1} x_1^2 \frac{e^{-2x_1} e^{-2x_2}}{\lambda e^{-\lambda x_1} \lambda e^{-\lambda x_2}} dx_2 dx_1 - \theta^2 \\ &= \frac{7}{4\lambda^2 (\lambda - 2)^4} - \theta^2 \end{aligned} \tag{2.18}$$

In order to find the optimal parameter for the minimum variance, the derivative of the variance with respect to λ should be calculated.

$$\frac{\partial var_\lambda (x)}{\partial \lambda} = \frac{-7(3\lambda - 2)}{2\lambda^3 (\lambda - 2)^5} = 0 \tag{2.19}$$

Then we have $\lambda^* = \frac{2}{3}$. Naïve Monte Carlo and IS results are given in Table 2.2 below which are ensured by our theoretical results.

Table 2.2. Results for the expectation of the maximum value of two exponential random variables

λ	$\hat{\theta}$	MSE	HL	LB	UB	AW	MinW	MaxW
Naïve Results								
1.00	1.50011	1.24969	2.19108	0.00000	3.69119			
IS Results								
0.40	1.50002	1.08725	3.23140	0.00000	4.73142	0.99965	0.00000	6.23812
0.50	1.50019	0.51429	1.98782	0.00000	3.48801	1.00096	0.00000	3.99622
0.60	1.49933	0.28092	1.34113	0.15820	2.84047	1.00071	0.00005	2.77499
0.61	1.50012	0.26977	1.30344	0.19669	2.80356	1.00007	0.00003	2.68645
0.62	1.49950	0.25997	1.26918	0.23031	2.76868	0.99953	0.00020	2.59948
0.63	1.49970	0.25365	1.24366	0.25604	2.74336	0.99949	0.00016	2.51812
0.64	1.50052	0.24820	1.22057	0.27995	2.72110	0.99933	0.00018	2.44062
0.65	1.49974	0.24420	1.20136	0.29837	2.70110	1.00016	0.00021	2.36555
0.66	1.49909	0.24259	1.18829	0.31080	2.68738	0.99944	0.00039	2.29493
0.67	1.49979	**0.24184**	1.17756	0.32224	2.67735	1.00067	0.00044	2.22616
0.68	1.50074	0.24319	1.17212	0.32862	2.67286	0.99986	0.00084	2.16063
0.69	1.49918	0.24620	1.17078	0.32840	2.66996	0.99985	0.00122	2.09987
0.70	1.49831	0.25084	1.17330	0.32501	2.67160	1.00057	0.00062	2.04032
0.80	1.50053	0.38695	1.36313	0.13740	2.86366	0.99974	0.01566	1.56177
0.90	1.50179	0.70260	1.73176	0.00000	3.23355	0.99971	0.15022	1.23427
1.00	1.50138	1.24941	2.19083	0.00000	3.69221	1.00000	0.99999	1.00000

We have a reduction factor of

$$\frac{var_{Naive}}{var_{IS}} = \frac{1.24969}{0.24184} = 5.17 \tag{2.20}$$

The 95 percent confidence interval for naïve Monte Carlo is

$$0.0 \leq \hat{\theta} \leq 3.69119 \tag{2.21}$$

whereas 95 percent confidence interval with IS is

$$0.32224 \leq \hat{\theta} \leq 2.67735 \tag{2.22}$$

We have the variance graph for example-2 in Figure 2.4. It is again *u-shaped* and we expect to have always *u-shaped* graphs with IS.

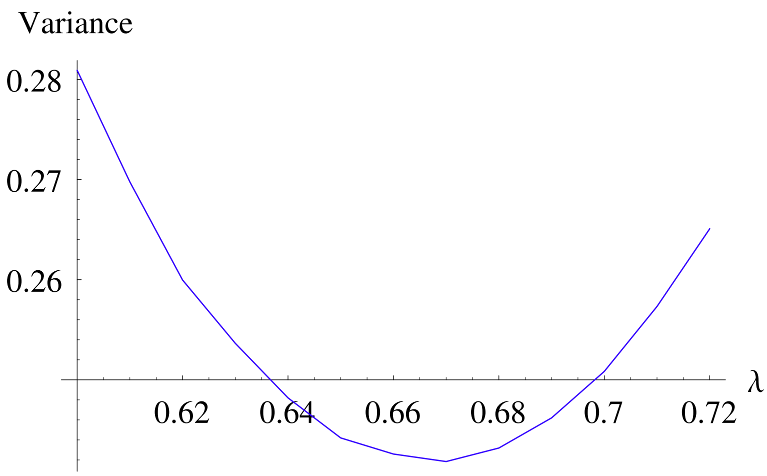


Figure 2.4. Variance graph for example 2

2.3.2. Pareto Importance Sampling Density

We used Pareto distribution as IS density for only service times because of having higher right tails. Although the original Pareto density has the form

$$f(x) = \frac{\alpha}{\beta}\left(\frac{\beta}{x}\right)^{\alpha+1}, x > 0 \tag{2.23}$$

where the parameters α and β are shape and scale parameters respectively, we used the shifted Pareto density:

$$f(x) = \frac{\alpha}{\beta}\left(\frac{\beta}{x+\beta}\right)^{\alpha+1}, x \geq 0, \alpha > 0, \beta > 0 \tag{2.24}$$

The term $(x+\beta)$ in the denominator provides to translate Pareto distribution where it now reflects the similar behavior with Exponential $\left(\lambda = \frac{\alpha}{\beta}\right)$. It is not exactly the same behavior, just similar that both of the pareto and Exponential have the same density value at $x = 0$ but Pareto has higher tails as can be seen in Figure 2.1.

2.3.3. Gamma Importance Sampling Density

We used Gamma IS density for interarrival times because it samples more in the important region just for interarrivals as can be seen in Figure 2.2. Gamma distribution with $0 < \alpha < 1$ does not have a closed form to generate samples by inversion. The acceptance-rejection algorithm in Law & Kelton's [16] book can be used. The Gamma distribution has the following density

$$f(x) = \begin{cases} \frac{\beta^{-\alpha} x^{\alpha-1} e^{-\frac{x}{\beta}}}{\Gamma(\alpha)} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.25}$$

Gamma reflects the same behavior with Exponential(λ) for $\alpha = 1$ and $\beta = \frac{1}{\lambda}$. We used Gamma distribution in $0.5 < \alpha < 1$ as Gamma has similar shape with Exponential for this interval of α.

2.3.4. Defensive Mixture Importance Sampling Densities

The IS densities of defensive mixture distributions are proposed to make IS robust. When a density which is not close to the original one is used as IS density, the danger of unsampled regions may occur leading to unbounded variances. The defensive mixture distributions prevent underrepresentation in some parts of the sample space. Defensive mixture distributions can be written as:

$$g_p(\mathbf{x}) = pf(\mathbf{x}) + (1-p)\, g_0(\mathbf{x}), \tag{2.26}$$

where $0<p<1$ and g_0 is an IS density, Hesterberg[1]. The term *defensive* refers to the use of the original density as one component of the mixture. This approach bounds the weight:

$$W(\mathbf{x}) = \frac{f(\mathbf{x})}{g_p(\mathbf{x})} \leq \frac{f(\mathbf{x})}{pf(\mathbf{x})} = \frac{1}{p}, \tag{2.27}$$

Thus it is guarenteed to have maximum weight of $\frac{1}{p}$, Hesterberg [1].

Choice of p:

The use of defensive mixture distributions is like combining IS with naïve simulation. The performance of this combination strongly depends on the parameters of the distributions rather than on the value of p which is the percentage of the original distribution in the defensive mixture distribution.

One advantage of defensive mixture distributions is that g_0 in equation (2.26) can be selected freely and a distribution which is easy to generate samples can be used. However this freedom can be limited with the high values of p when the sampling region does not fit well with g_0. Another advantage of defensive mixture distributions is the limitation of the weights of IS by $(\frac{1}{p})^d$ where d is the dimension of the integral. Thus p determines the maximum weight. In general it is impossible to control the sampling process totaly. That is why defensive mixture distributions have a basic role in comparatively complex poblems.

The choice of p depends on the complexity of the problem and can be optimized by a pilot study. It is obvious that if $p = 0$ we do not have a mixture IS density and if $p = 1$ we have the original naïve experiment.

Mixture Distributions:

In some cases the original distribution may be unknown or it may be impossible to generate samples. Then we cannot use the efficient defensive mixture distributions like in the form of equation (2.26). If we remove the original density in equation (2.26) and the term defensive in the name, we get the general mixture distributions

$$g\left(\mathbf{x}\right) = \sum_{i=1}^{I} p_i g_i\left(\mathbf{x}\right), \tag{2.28}$$

where $\sum_{i=1}^{I} p_i = 1$ and $p_i > 0, \forall i$. This form of IS densities may represent the sampling region better if the original density $f(x)$ is difficult to approximate.

3. INDEPENDENT IMPORTANCE SAMPLING FOR M/M/1 QUEUES

3.1. Independent Importance Sampling Densities

Now we investigate good IS densities to find the expectation of a buffer overflow in an M/M/1 queueing system. We have an indicator $q(\mathbf{X})$ function

$$q(\mathbf{X}) = \begin{cases} 1 & \text{if buffer is exceeded during simulation} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

We implemented the IS method in both C and ARENA softwares for this thesis. Arena was selected because of the easy modelling and implementation features, and C was selected because of its runtime advantage (See Chapter 5).

We consider a simple M/M/1 queueing system with a limited buffer as variance reduction is a difficult job for especially complex systems. Essentially it is important to find close to optimal IS densities for rare events. That is why we carry out IS on a simple M/M/1 queueing system for satisfactory results. Because of the Markovian property, we originally have Exponential interarrival and service times. For simplicity the service times distribution is fixed to Exponential($\nu = 1$) and we used different Exponential(λ) interarrival times distributions so that the traffic intensity ρ is equal to λ. In order to have a small probability for a buffer overflow, we select small values of λ but increase it with the increase of the number of customers allowed to enter the system. We had focused on the application of IS for transient M/M/1 queueing system, so the steady state requirement of the "the rate of interarrival times should be smaller than the rate of service times" is not a must,[17]. It is possible to freely select the parameters of the IS densities for the importance of the optimization of IS. We experimented with ten systems given in Table (3.1).

Table 3.1. Observed M/M/1 systems

	Number of Arrivals	Buffer Size	1	2
1	2	0	ρ=0.1	ρ=0.2
2	3	1	ρ=0.1	ρ=0.2
3	11	5	ρ=0.7	ρ=0.8
4	101	20	ρ=0.7	ρ=0.8
5	101	22	ρ=0.7	ρ=0.8

We use IS densities in two version: The first one is for interarrival times and the second one is for service times. For instance, exponential interarrival times with higher rates and exponential service times with smaller rates allow to make buffer overflow more frequent. We have the original naïve integral

$$\theta = E_f\left[q\left(\mathbf{X}\right)\right] = \int_{R^d} q\left(\mathbf{X}\right) f\left(\mathbf{x}\right) d\mathbf{x} \tag{3.2}$$

where $\mathbf{X}$ is a random vector consisting of interarrival times and service times. The joint distribution of $\mathbf{X}$ can be written as

$$f(\mathbf{x}) = \prod_{i=1}^{A-1} f_a\left(x_i\right) \prod_{j=1}^{S} f_s\left(x_j\right) \tag{3.3}$$

where notations f_a and f_s indicate the interarrival times distribution and service times disribution, respectively. A is the number of customers allowed to enter the system. And S is the number of service times. In practice $A = S$ and we have $A-1$ interarrivals. In this chapter like in the previous we use Independent Importance Sampling (IIS) where there is no dependence between consecutive interarrival times and consecutive service times and between any interarrival time and any service time. We generated iid random variables using Law & Kelton's [16] uniform generator that is included to our C program. To make things easier we first had simulation runs in a large parameter range with big increments of the parameter values and then tried to find optimal parameters in a second experiment for a smaller parameter interval found in the first experiment.

When an IS density is used for just interarrivals or for just service times, the expectation θ can be written as

$$\theta = \overbrace{\int \cdots \int}^{dimension-d} q(\mathbf{X}) \prod_{i=1}^{d} \frac{f(x_i)}{g(x_i)} \prod_{i=1}^{d} g(x_i)\, dx_d dx_{d-1} \cdots dx_1 \tag{3.4}$$

where d is the dimension of the integral. In theory dimension is equal to the sum of the number of interarrivals and number of services. But in practise, d can be calculated as follows: Every arrival after the first one has a positive influence on buffer overflow, so we have $(A-1)$ dimension from interarrivals and we have a flexibility of buffer size trying to prevent any quitings that adds $(S-BS)$ dimension and finally last customer's service time does not influence the buffer overflow that we should substract it from the total dimension. Briefly d is calculated

Number of Interarrivals	$:A-1$
Number of Services	$:S-BS-1$
In practise	$:A=S$

$$d = [2(A-1) - \mathrm{BS}]\,. \tag{3.5}$$

We used Exponential IS densities with different rates. We used Pareto distribution for just service times as in equation (2.24) to have better results. Pareto has higher tails than Exponential, so we cannot use it for interarrivals as that does not make buffer overflow more frequent. We used Gamma IS density with $0.5 \leq \alpha < 1$ that the Gamma distribution is similar to Exponential(λ) for ($\alpha = 1$ and $\beta = 1/\lambda$). And we used defensive mixture of Exponential and Uniform densities and also defensive mixture of Exponential and Gamma densities for $0 \leq p \leq 1$. As Pareto and Gamma densities have two parameters, we tried to optimize both α and β and had to run many experiments for that task.

3.2. Results for 2 Arrivals and Buffer Size of 0

This example is to get a first idea about how IS can be applied to M/M/1 queues. In the next sections we will look at systems with higher number of arrivals. We have two dimensional integral of equation (3.4) for the expectation of a buffer overflow for this case. That is because the interarrival time and the first service time together determine any buffer overflow.

The results are displayed step by step for different IS densities. The results for just Exponential interarrival times IS density for $\rho = 0.1$ are displayed in Table 3.2 and for other IS densities we put the results to Appendix B. The naïve results are displayed in the first two lines of the tables for both C and ARENA programs with 10^6 and 10^5 replications respectively. In the latter parts of Tables just the IS results obtained with C are displayed. The abbreviations in the Table headings can be found in the List of Symbols / Abbreviations page.

For the best parameter $\lambda_{IS} = 0.62$ we obtained a variance reduction factor of

$$\frac{Var_{naive}}{Var_{IS}} = \frac{0.08249}{0.01919} = 4.30 \tag{3.6}$$

And the 95 percent confidence intervals for naïve estimator and $IS_{\lambda=0.62}$ are

$$0.09015 \leq \hat{\theta}_{naive} \leq 0.09128 \tag{3.7}$$

$$0.09040 \leq \hat{\theta}_{IS} \leq 0.09094 \tag{3.8}$$

respectively, where IS confidence interval includes the naïve estimator. Moreover, we have average weights close to 1 which indicates that the results are reliable. The variance graph for this case with $\rho = 0.1$ is shown in Figure 3.1 and it just looks like the expected one shown in section 2.3.1.

As the ARENA software has a longer run time with respect to C, we give the IS

Table 3.2. Results for Exponential IS Density for 2A0B and $\rho = 0.1$

λ	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
Naïve									
0.10	1000000	0.09071	0.08249	0.00056	0.09015	0.09128	**C**		
0.10	100000	0.09042	0.08248	0.00178	0.08864	0.09220	**ARENA**		
IS									
0.15	1000000	0.09106	0.05530	0.00046	0.09060	0.09152	1.00017	0.66667	99.8
0.20	1000000	0.09115	0.04185	0.00040	0.09075	0.09155	1.00203	0.50000	2048.0
0.25	1000000	0.09129	0.03396	0.00036	0.09093	0.09165	0.99999	0.40000	3292.4
0.30	1000000	0.09105	0.02882	0.00033	0.09072	0.09138	0.99978	0.33333	7471.0
0.35	1000000	0.09095	0.02535	0.00031	0.09064	0.09126	1.03901	0.28571	41347.2
0.40	1000000	0.09101	0.02297	0.00030	0.09071	0.09131	1.06992	0.25000	65536.0
0.45	1000000	0.09094	0.02142	0.00029	0.09065	0.09123	0.97990	0.22222	26446.7
0.50	1000000	0.09084	0.02026	0.00028	0.09057	0.09112	0.99184	0.20000	33237.6
0.55	1000000	0.09092	0.01968	0.00027	0.09064	0.09119	0.99592	0.18182	39708.8
0.60	1000000	0.09081	0.01921	0.00027	0.09055	0.09109	1.00099	0.16667	45706.1
0.61	1000000	0.09071	0.01924	0.00027	0.09040	0.09098	0.99869	0.16394	46840.9
0.62	1000000	0.09067	**0.01919**	0.00027	0.09040	0.09094	0.99156	0.16129	47953.2
0.63	1000000	0.09072	0.01920	0.00027	0.09045	0.09099	1.00139	0.15873	49042.9
0.64	1000000	0.09078	0.01924	0.00027	0.09051	0.09105	1.00200	0.15625	50109.6
0.65	1000000	0.09097	0.01926	0.00027	0.09070	0.09125	1.01810	0.15385	51153.5
0.70	1000000	0.09109	0.02015	0.00028	0.09081	0.09137	1.01267	0.14286	56027.6
0.75	1000000	0.09097	0.02137	0.00029	0.09068	0.09125	0.93072	0.13333	60337.7
0.80	1000000	0.09104	0.02252	0.00029	0.09075	0.09133	1.02696	0.12500	64112.3
0.85	1000000	0.09106	0.02345	0.00030	0.09076	0.09136	0.98141	0.11765	67390.1
0.90	1000000	0.09106	0.02540	0.00031	0.09075	0.09137	0.97142	0.11111	70214.1
0.95	1000000	0.09105	0.02640	0.00032	0.09073	0.09137	0.95009	0.10526	72628.2

results with ARENA only for optimal $\lambda^* = 0.62$.

Table 3.3. Results with ARENA for Exponential IS Density for 2A0B and $\rho = 0.1$

λ_{IS}	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
0.62	100000	0.09044	0.01866	0.00085	0.08959	0.09129	1.0505	0.16129	20111

Although ARENA gives a smaller variance estimator than C, the confidence interval with ARENA is wider than with C because of the smaller sample size. Now we

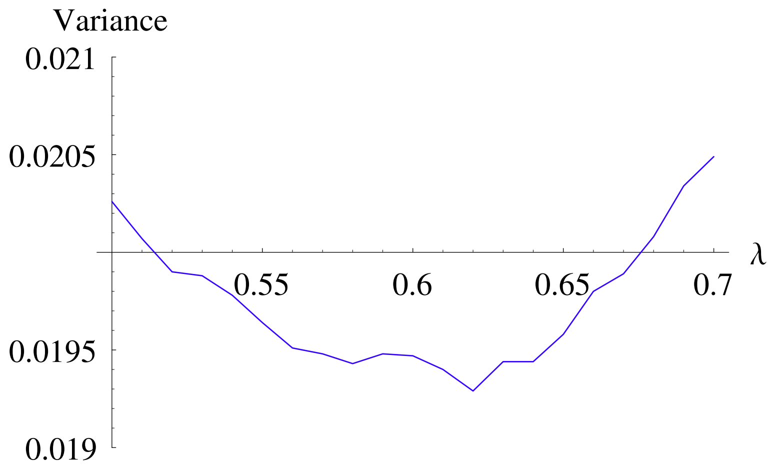


Figure 3.1. Variance graph of Exponential IS density for 2A0B and $\rho = 0.1$

can search for other IS densities.

The results for Pareto IS density are shown in Appendix B in Table B.2. We have a variance reduction factor of

$$\frac{Var_{naive}}{Var_{IS}} = \frac{0.08249}{0.07092} = 1.16 \tag{3.9}$$

for the best parameters $\alpha = 18$, $\beta = 2.5$ which is not satisfactory, and 95 percent confidence interval is

$$0.08980 \leq \hat{\theta} \leq 0.09084 \tag{3.10}$$

for $IS_{(\alpha=18,\beta=2.5)}$ that includes the naïve estimator. Moreover, we have average weight close to 1 which indicates that the results are reliable. And variance graph is shown in Figure 3.2.

As discussed in section (2.3.4) defensive mixture distributions lead to high performance when used as IS densities. We used defensive mixture of Exponential and Gamma densities for interarrival times. The effect of Gamma interarrivals can be seen in the Figure 2.2. The results displayed in Table B.3 show that for optimal results we do not have a mixture as we have the optimal results for $p = 0.0$. The results of

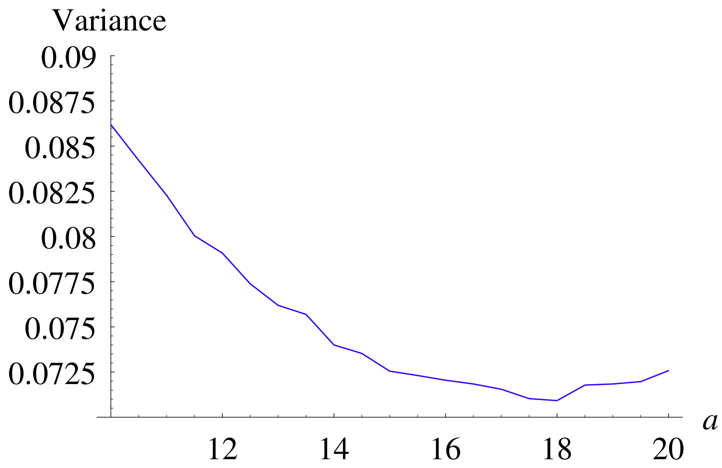


Figure 3.2. Variance graph of Pareto IS density for $\beta = 2.5$

defensive mixture of Exponential and Gamma densities for some other values of p and α are shown in Table B.5 in Appendix B. The factor of variance reduction is

$$\frac{Var_{naive}}{Var_{IS}} = \frac{0.08249}{0.01930} = 4.27 \tag{3.11}$$

which is close to the Exponential IS density. The 95 percent confidence interval is

$$0.09051 \leq \hat{\theta} \leq 0.09106 \tag{3.12}$$

which includes the naïve estimator. And the variance graph of defensive mixture of Exponential and Gamma IS density for $p = 0$ and $\alpha = 0.9$ is shown in Figure 3.3 that we have the best variance for $\beta = 1.7$.

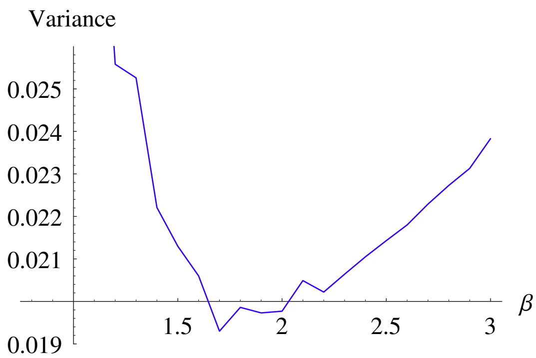


Figure 3.3. Variance graph of defensive mixture of Exponential and Gamma IS density for p=0 and $\alpha = 0.9$

We used another defensive mixture distribution consisting of original Exponential and Uniform densities. The Uniform density is in the following form,

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{if } a < x < b \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

The results are displayed in Table B.4 for $p = 0.3$ that provided the best result. We have $\hat{\theta}_{(b=3.2)} = 0.09067$ and $MSE^*_{(b=3.2)} = 0.03310$. The factor of variance reduction is

$$\frac{Var_{naive}}{Var_{IS}} = \frac{0.08249}{0.03310} = 2.5 \tag{3.14}$$

and the 95 percent confidence interval is

$$0.09032 \leq \hat{\theta} \leq 0.09103 \tag{3.15}$$

Although defensive mixture of Exponential and Uniform could not give the best results, it is interesting to see the value of maximum weights in the defensive mixture distributions. And the variance graph of defensive mixture distribution for $p = 0.3$ and varying b is shown in Figure 3.4.

Consequently, when we take into account all IS densities, Exponential IS density with changing the rate gives the best performance for two arrivals case. It leads to the highest factor of variance reduction with a factor of 4.30. It is also the easiest density to draw samples that provides the shortest run time.

3.3. Results for 3 Arrivals and Buffer Size of 1

We have a three dimensional integral for this case where two out of them come from interarrivals and one comes from the first service time. Here we give all optimal

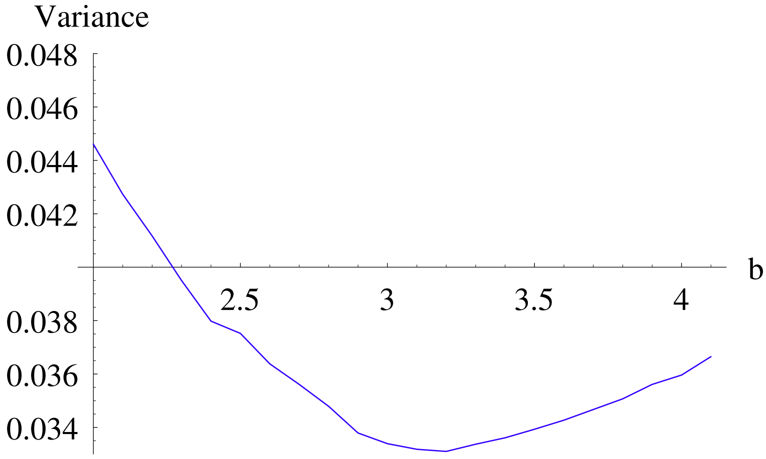


Figure 3.4. Variance graph of defensive mixture of Exponential and Uniform IS density

results for different IS densities in Table 3.4 and put all results with different parameters into the Electronic Appendix. The results are similar to the two arrivals case but the variance reduction is larger.

Table 3.4. Optimal Results for 3A1B and $\rho = 0.1$

	$\hat{\theta}$	MSE	LB	UB	Factor
Naïve C, λ=0.1	0.00820	0.00813	0.00802	0.00837	
Naïve ARENA, λ=0.1	0.00809	0.00793	0.00754	0.00864	
IS Exponential ($\lambda = 0.67$)	0.00825	0.00066	0.00820	0.00830	12.31
IS Pareto($\alpha = 16.5, \beta = 3$)	0.00817	0.00513	0.00803	0.00831	1.59
IS Def-Mix Exp&Uniform ($b = 1.7$), p=0.4	0.00828	0.00174	0.00820	0.00836	4.68
IS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 2$),p=0.0	0.00830	0.00073	0.00825	0.00836	11.14

We have the factors of variance reduction at the last column of Table 3.4 for different IS densities. We have the largest variance reduction and smallest confidence interval by only changing the rate of Exponential density. We have $p = 0.4$ for optimal results with defensive mixture of Exponential and Uniform IS density. And we do not have any mixture for Gamma IS density as the optimal result occurs for $p = 0.0$.

3.4. Results for 11 Arrivals and Buffer Size of 5

We have a fifteen dimensional integral for this case where ten dimensions come from interarrivals and five come from the service times. Here we give all optimal results for different IS densities in Table 3.5. From now on we will display all results in this way. However full results can be found in the electronic Appendix. The results are interesting. We have smaller variance reduction compared to the previous cases. One reason for that is the dimension of the problem. The factor of variance reduction decreases when the dimension increases.

Table 3.5. Optimal Results for 11A5B and $\rho = 0.7$

	$\hat{\theta}$	MSE	LB	UB	Factor
Naïve C, λ=0.7	0.05143	0.04879	0.05100	0.05186	
Naïve ARENA, λ=0.7	0.05193	0.04957	0.05055	0.05331	
IS Exponential ($\lambda = 0.87$)	0.05131	0.03020	0.05097	0.05166	1.62
IS Pareto($\alpha = 19.5, \beta = 17.5$)	0.05089	0.03020	0.05055	0.05123	1.62
IS Def-Mix Exp&Uniform ($b = 1.0$), p=0.8	0.05125	0.03705	0.05087	0.05163	1.32
IS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 0.9$),p=0.6	0.05142	0.02958	0.05109	0.05176	1.65

Another reason for the small variance reduction factor is the estimator itself. We have a comperatively larger estimator than for the three arrivals case. It means that we have larger number of samples that the buffer has overflowed. For the naïve Monte Carlo we count the number of occurences of rare events and divide it the sample size. So we have a larger number of points to apply IS and this reduces the variance reduction factor. Thus we can conclude that IS performs better for very rare events.

3.5. Results for 101 Arrivals and Buffer Size of 20

We have a hundredeighty dimensional integral for this case where hundred dimensions come from interarrivals and eighty come from the service times. Although we can calculate even thousand dimensional expectations via naïve simulations, it is really a hard job to have a variance reduction with high dimensional problems. All optimal results for different IS densities are shown in Table 3.6.

Table 3.6. Optimal Results for 101A20B and $\rho = 0.7$

	$\hat{\theta}$	MSE	LB	UB	Factor
Naïve C, λ=0.7	0.00265	0.00264	0.00255	0.00275	
Naïve ARENA, λ=0.7	0.00263	0.00234	0.00233	0.00293	
IS Exponential($\lambda = 0.76$)	0.00260	0.00127	0.00253	0.00267	2.09
IS Pareto($\alpha = 28.5, \beta = 21.5$)	0.00260	0.00126	0.00253	0.00267	2.09
IS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 0.9$),p=0.8	0.00262	0.00124	0.00255	0.00269	2.13

We do not have sensible results with defensive mixture of Exponential and Uniform IS densities that is why we do not show results for it. And as expected, we have optimal result for an higher value of $p = 0.8$ for defensive mixture of Exponential and Gamma IS density for 101 arrivals. We show the behavior of variance for different values of p for defensive mixture of Exponential and Gamma IS density in Figure 3.5.

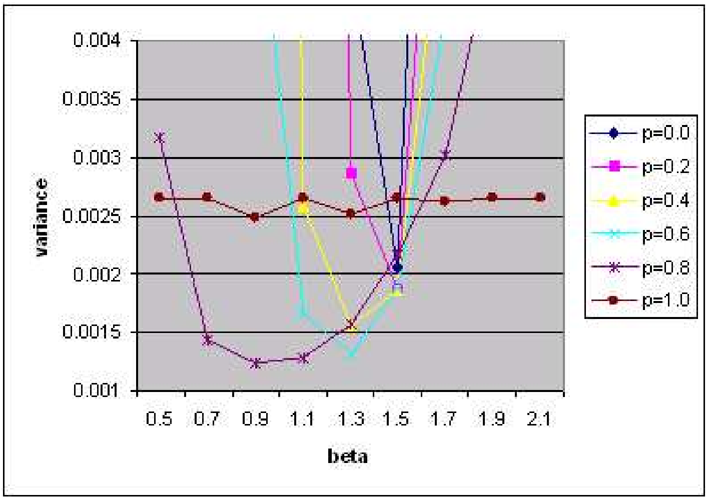


Figure 3.5. Variance graph for different p values of defensive mixture of Exponential and Gamma IS density

3.6. Results for 101 Arrivals and Buffer Size of 22

We have a hundredseventyeight dimensional integral for this case where hundred dimensions come from interarrivals and seventyeight come from the service times. We consider to increase the buffer size in order to decrease the probability of buffer overflow and consequently supply higher variance reduction factor. All optimal results for different IS densities are shown in Table 3.7.

Table 3.7. Optimal Results for 101A22B and $\rho = 0.7$

	$\hat{\theta}$	MSE	LB	UB	Factor
Naïve C, λ=0.7	0.001140	0.001139	0.001074	0.001206	
Naïve ARENA, λ=0.7	0.001180	0.001179	0.000879	0.001481	
IS Exponential($\lambda = 0.8$)	0.001135	0.000468	0.001093	0.001178	2.43
IS Pareto($\alpha = 29$, $\beta = 22$)	0.001117	0.000431	0.001077	0.001158	2.64
IS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 1.2$),p=0.6	0.001159	0.000502	0.002525	0.002622	2.27

We have optimal result for p=0.6 for defensive mixture of Exponential and Gamma IS density as shown in Table 3.7 that defensive mixture IS distributions are very useful for higher dimensional problems. When we compare the results with hundredone arrivals and buffer size of twenty case we have higher factor of variance reduction. The reason for this is that we have less frequent probability of buffer overflow.

4. DEPENDENT IMPORTANCE SAMPLING FOR M/M/1 QUEUES

In this Chapter we propose Dependent Importance Sampling (DIS). Either service times are calculated depending on the value of the interarrival times or vice versa (interarrival times dependening on the service times). We organised the sections here as in Chapter 3 and give results for different IS densities in accordance. We have two sections in this chapter, the first one presents the IS densities for dependent service times and the second one explains a new idea for sampling from the important region. As we mentioned in section 2.2 IS performs well for rare events occuring in the tails of distributions. Our dependent interarrival times experiments did not show sensible results except for 2 dimensions. So we do not include any results with dependent interarrival times here.

4.1. Dependent Service Times

We used the idea that the service times should be greater than interarrival times to increase the probability of buffer overflows. We first generated interarrival times. Then we generated service times depending on the interarrival times so that the service times could be greater than the interarrival times. Thus, we collected pairs of dependent random variables consisting of first interarrival time with first service time and second interarrival time with second service time and so on. Because of this dependence, we have a bivariate weight function consisting of pairs of interarrival and service times. Essentially we do not have any dependence between pairs in this section.

In order to have service times often greater than the corresponding interarrival times, it is obvious that some positive additional term should be added to interarrival times and then service times should be generated with using the new term. This new term becomes the parameter of the dependent IS density. We use the mean values of independent IS densities to find the parameters of dependent IS densities. Algorithm

4.1 below describes the process. Let us first define some terms:

X	:interarrival time, Exponential(λ)
Y	:service time, Exponential(ν)
IS densities	:Exponential(ν'), Defensive Mixture of Exponential & Gamma(α, β'), Defensive Mixture of Exponential & Uniform($0, b'$)
c	:any nonnegative real number $c > 0$

Algorithm 1 Generating dependent service times from IS densities

1. Generate X from Exponential(λ)
2. Generate Y from Exponential(ν') or Defensive Mixture of Exponential and Gamma (α, β') or Defensive Mixture of Exponential and Uniform($0, b'$). All ν', β' and b' can be calculated seperately as in the following

$$E\left[Y|X\right] = \frac{1}{\nu} + cX, \Rightarrow \nu' = \frac{1}{\frac{1}{\nu} + cX} = \frac{\nu}{1 + \nu cX} \tag{4.1}$$

$$E\left[Y|X\right] = \alpha\beta' = \alpha\beta + cX, \Rightarrow \beta' = \beta + \frac{cX}{\alpha} \tag{4.2}$$

$$E\left[Y|X\right] = \frac{b'}{2} = \frac{b}{2} + cX, \Rightarrow b' = b + 2cX \tag{4.3}$$

3. Calculate weight as $w(X, Y) = \frac{f(X,Y)}{g(X,Y)}$
4. Calculate $\hat{\theta}$ and MSE.

We add some part of the interarrival time X to the mean service times of different IS densities as seen in equations 4.1, 4.2 and 4.3. Then we have continuosly changing parameters of IS densities. We use c in $(0.1 < c < 1.0)$ for two and three arrivals and in $(0.01 < c < 0.1)$ for 11 and 101 arrivals. It is hard to find optimal solutions in high dimensional problems so we can add just a small term for dependence. Although we have a dependency, the results are reliable because we use the correct weight functions.

4.1.1. Results for 2 Arrivals and Buffer Size of 0

We have more satisfactory results than for independent IS densities. The most important point is that the estimator $\hat{\theta}$ is not biased because of the dependence. But surely we have sensible results for a narrower parameter range of IS densities. The results are shown in Table 4.1. Although the probability of buffer overflow is comperatively higher for this case, we have good factor of variance reduction for each IS density. For defensive mixture distributions we have optimal results for $p = 0.0$ that we may expect to have higher p values for higher dimensional cases. We use c values in $0.1 < c < 1$ and the optimal results occur for $c = 0.7$ as can be seen in Table 4.1 below.

Table 4.1. Optimal Results with dependent IS densities for 2A0B and $\rho = 0.1$

	$\hat{\theta}$	MSE	LB	UB	Factor
Naïve C, λ=0.1	0.09071	0.08249	0.09015	0.09128	
Naïve ARENA, λ=0.1	0.09042	0.08248	0.08864	0.09220	
DIS Exponential($\lambda = 0.9$),c=0.7	0.09087	0.00810	0.09069	0.09105	10.18
DIS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 1.2$)	0.09093	0.00915	0.09074	0.09112	9.02
IS Def-Mix Dept&Indept Exp($\lambda = 0.9$),p=0,c=0.7	0.09094	0.00813	0.09076	0.09112	10.14

One contribution of this thesis is using defensive mixture of independent and dependent IS densities. For defensive mixture distributions, as we usually do, we generate a Uniform$(0, 1)$ random variable and check it with the p value. If the random variable is smaller than p, we have independent IS, otherwise we have dependent IS. However we have $p = 0$ for two arrivals. This indicates that dependent IS outferforms independent IS for dimension two. But we can surely expect higher p values for higher dimensions because when we sample more, the effect of the dependence may be violated as we do not have any dependence between pairs.

We have so close results for this case. The very little difference is probabily caused from the diferent seeds that we use for the replications. Some detailed results are shown in Table 4.2. Because of the dependence, maximum weights are increaseing.

We put the results with $c = 2$ to the Appendix C in Table C.1 to have an idea about the c values. The results are so close to the ones with c between 0.1 and 1.0.

Table 4.2. Results for dependent Exponential IS density with $c = 1$, 2A0B $\rho = 0.1$

λ	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
0.1	1000000	0.09121	0.06626	0.00050	0.09071	0.09172	0.99931	0.0	78.6
0.2	1000000	0.09100	0.03147	0.00035	0.09065	0.09134	0.98552	0.0	2094.0
0.3	1000000	0.09089	0.02018	0.00028	0.09062	0.09117	0.97016	0.0	6403.9
0.4	1000000	0.09096	0.01477	0.00024	0.09072	0.09119	0.94468	0.0	9754.9
0.5	1000000	0.09102	0.01175	0.00021	0.09081	0.09124	0.95222	0.0	15941.2
0.6	1000000	0.09104	0.00994	0.00020	0.09084	0.09123	0.93127	0.0	24736.5
0.7	1000000	0.09106	0.00888	0.00018	0.09088	0.09125	0.88688	0.0	32341.8
0.8	1000000	0.09099	0.00836	0.00018	0.09081	0.09117	0.86865	0.0	38221.3
0.9	1000000	0.09088	**0.00824**	0.00018	0.09071	0.09106	0.75323	0.0	38100.8
1.0	1000000	0.09090	0.00851	0.00018	0.09072	0.09108	0.72142	0.0	42285.2
1.1	1000000	0.09091	0.00918	0.00019	0.09072	0.09110	0.68551	0.0	45273.8
1.2	1000000	0.09097	0.01032	0.00020	0.09077	0.09117	0.66904	0.0	47256.7
1.3	1000000	0.09097	0.01212	0.00022	0.09075	0.09119	0.64976	0.0	48427.3
1.4	1000000	0.09094	0.01469	0.00024	0.09071	0.09118	0.71933	0.0	48959.0

And variance graph of dependent Exponential IS density is shown in Figure 4.1 that it looks like the independent case. As we can see from the figure, the variance has a local minima for $\lambda = 0.9$.

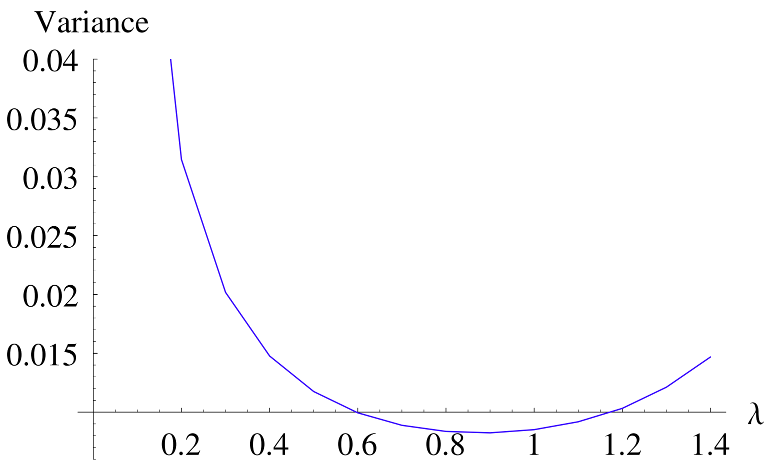


Figure 4.1. Variance graph of dependent Exponential IS density

4.1.2. Results for 3 Arrivals and Buffer Size of 1

We have very good results for the three arrivals case. The results are shown in Table 4.3. We have almost 29 times variance reduction with the new idea of defensive mixture of dependent and independent Exponential IS densities without violating the estimator $\hat{\theta}$. The low dimension and small probability of buffer overflow leads to the good results.

Table 4.3. Optimal Results with dependent IS densities for 3A1B and $\rho = 0.1$

	$\hat{\theta}$	MSE	LB	UB	Factor
Naïve C, λ=0.1	0.00819	0.00813	0.00802	0.00837	
Naïve ARENA, λ=0.1	0.00809	0.00793	0.00754	0.00864	
DIS Exponential($\lambda = 0.8$),c=0.5	0.00824	0.00032	0.00820	0.00827	25.75
DIS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 1.4$)	0.00829	0.00028	0.00826	0.00832	28.84
IS Def-Mix Dept&Indept Exp($\lambda = 0.8$),p=0,c=0.5	0.00826	0.00032	0.00823	0.00830	25.08

For defensive mixture of Exponential and dependent Gamma IS density we have optimal results for $p = 0.0$ and $c = 0.5$. Therefore the comments for two arrivals case are also valid for three arrivals case.

4.1.3. Results for 11 Arrivals and Buffer Size of 5

The results are shown in Table 4.4. Although we do not have very satisfactory factor of variance reductions, they are better than the independent case of 11 arrivals. We have optimal result for dependent Exponential($\lambda = 0.88$) where we had optimal result for independent Exponential($\lambda = 0.87$). This small increment in the rate is sensible because the dependence tends to increase the buffer overflow probability just like the changing the rate of Exponential IS density.

We do not have a defensive mixture for dependent Gamma IS density as the optimal results occurs for $p = 0.0$ and $c = 0.01$ but for defensive mixture of independent

Table 4.4. Optimal Results with dependent IS densities for 11A5B and $\rho = 0.7$

	$\hat{\theta}$	MSE	LB	UB	Factor
Naïve C, λ=0.7	0.05143	0.04879	0.05100	0.05186	
Naïve ARENA, λ=0.7	0.05193	0.04957	0.05055	0.05331	
DIS Exponential($\lambda = 0.88$),c=0.09	0.05146	0.02693	0.05114	0.05178	1.81
DIS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 1.3$)	0.05146	0.01984	0.05119	0.05174	2.50
IS Def-Mix Dept&Indept Exp($\lambda = 1.2$) p=0.4,c=0.09	0.05143	0.02445	0.05112	0.05173	2.00

and dependent Exponential IS density we have $p = 0.4$ that clearly shows this idea is very useful.

4.1.4. Results for 101 Arrivals and Buffer Size of 20

We have better results than for the eleven arrivals case. It is simply because of the estimator $\hat{\theta}_{11A}$ is nearly 20 times greater than $\hat{\theta}_{101A}$. The results are shown in Table 4.5. For 180 dimensions we can conclude to have very satisfactory results. We have optimal result for dependent Exponential($\lambda = 0.80$) where we had optimal result for independent Exponential($\lambda = 0.76$). We have a defensive mixture not only for dependent Gamma IS density where the optimal results occurs for $p = 0.6$ and $c = 0.09$ but also for defensive mixture of independent and dependent Exponential IS density where we have $p = 0.4$. This clearly shows that defensive mixture distributions are very useful for higher dimensional problems.

Table 4.5. Optimal Results with dependent IS densities for 101A20B and $\rho = 0.7$

	$\hat{\theta}$	MSE	LB	UB	Factor
Naïve C, λ=0.7	0.00265	0.00264	0.00255	0.00275	
Naïve ARENA, λ=0.7	0.00263	0.00234	0.00233	0.00293	
DIS Exponential($\lambda = 0.8$),c=0.03	0.00266	0.00129	0.00259	0.00273	2.05
DIS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 1.3$)	0.00257	0.00062	0.00252	0.00262	4.28
IS Def-Mix Dept&Indept Exp($\lambda = 1.0$) p=0.6,c=0.09	0.00256	0.00085	0.00250	0.00262	2.74

4.1.5. Results for 101 Arrivals and Buffer Size of 22

We have a hundredseventyeight dimensional integral for this case where hundred dimensions come from interarrivals and seventyeight come from the service times. We consider to increase the buffer size in order to decrease the probability of buffer overflow and consequently supply higher variance reduction factor. All optimal results for different IS densities are shown in Table 4.6.

Table 4.6. Optimal Results with IS densities for 101A22B and $\rho = 0.7$

	$\hat{\theta}$	MSE	LB	UB	Factor
Naïve C, λ=0.7	0.001140	0.001139	0.001074	0.001206	
Naïve ARENA, λ=0.7	0.001210	0.001049	0.000974	0.001246	
DIS Exponential($\lambda = 0.8$)	0.001164	0.000405	0.001125	0.001204	2.81
DIS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 1.1$)	0.001116	0.000315	0.002525	0.002622	3.61
IS Def-Mix Dept&Indept Exp($\lambda = 0.9$),p=0.5	0.001136	0.000301	0.001081	0.001151	3.78

We have optimal result for $c = 0.09$ and $p = 0.6$ for defensive mixture of Exponential and Gamma IS density and $c = 0.07$ for defensive mixture of independent and dependent Exponential IS densities. When we compare the results with both independent case and dependent hundredone arrivals and buffer size of twenty case we have higher factor of variance reduction. It is clearly because of the dependence and the less frequent probability of buffer overflow.

4.2. Cut Method

We propose a new approach for calculating the expectations via dependent random variables. We again use dependent service times in this section. We call this new technique *Cut Method* as we cut the sampling plane so that we just sample important vectors. These important vectors are consisting of interarrival times and service times that are greater than the interarrival times. And we have a conditional joint IS denity for the cut method.

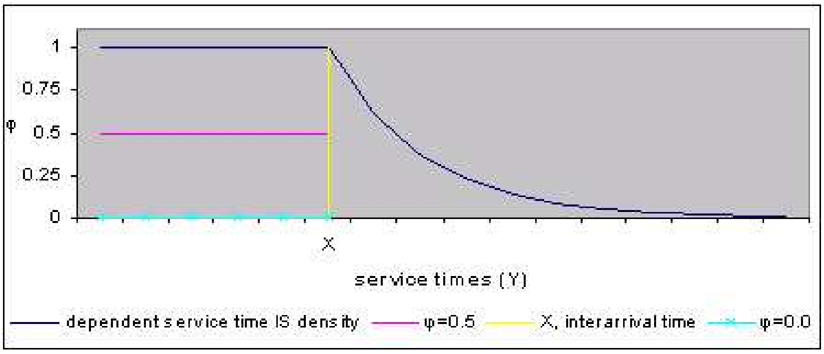


Figure 4.2. Application of cut method with known interarrival time X

We can reach zero variance (see Chapter 2 and section 2.2 for details) if we can find the correct marginal distribution. The marginal distribution turns out to be Exponential for our example. Then it will be enough to find the correct parameters of the dependent joint IS density.

The application of this method is similar to the application of defensive mixture of Exponential and Uniform IS density. We have a probability called ψ to determine from where to sample. In Figure 4.2 we have the interarrival time X with the yellow line and it cuts the sampling region into two parts. The important region is the area below the blue curve in the right handside of the yellow line. This part of the graph is the graph of original service times density and as it is a pdf, the area is equal to one. And we can easily find the area of rectengular in the left handside of the yellow line. We have already set the original service times distribution to Exponential($\nu = 1$) and we suppose to have a zero service time for the point X in Figure 4.2. Then the cover point in the y-axis is 1 and we have $\varphi = 1$ for this point and the area becomes $A = \varphi\nu X$. We should minimize this area in order not to sample from there because it only increases the sampling effort. As seen in Figure 4.2 we can use small φ values and even can use $\varphi = 0$. Thus, pdf graph of service times IS density jumps at the point X. The full process with cut method is described in Algorithm 4.2. Firstly we should set the following definitions:

X	:Interarrival times, Exponential(λ),
Y	:Service times, Exponential(ν),
φ	:Factor for limiting the unimportant region,
ψ	:Probability to determine from where to sample.

Algorithm 2 Cut Method Sampling Algorithm

1. Generate X from independent Exponential(λ) IS density,
2. Calculate ψ as in the following,

$$\psi = \frac{\text{Area of the rectengular in Figure 4.2}}{\text{Total Area}} = \frac{\varphi X \nu e^{-\nu}}{1 + \varphi X \nu e^{-\nu}} \tag{4.4}$$

   Note that for $\varphi = 0$ $\psi = 0$ as well.

3. Generate random variable u from Uniform(0,1).
4. Determine the sampling region,

   if(u $\leq \psi$) generate service time from Uniform(0,X)

   else generate service time as Y=X+ExpoRand(ν)

5. Calculate conditional service time IS density for Y,

$$g\left(Y|X\right) = \begin{cases} \frac{\varphi \nu e^{-\nu}}{1+\varphi X \nu e^{-\nu}} & \text{if } Y < X \\ \frac{\nu e^{-\nu(Y-X)}}{1+\varphi X \nu e^{-\nu}} & \text{otherwise} \end{cases} \tag{4.5}$$

6. Calculate weight as $w(X,Y) = \frac{f(X,Y)}{g(X,Y)}$
7. Calculate $\hat{\theta}$ and MSE.

It is easy to see that we have optimal results for $\varphi = 0$ and it also leads to a run time advantage.

4.2.1. Results for 2 Arrivals and Buffer Size of Zero

We show the results with conditional IS density for two arrivals in Table 4.7. The results are very good. We have a zero variance estimator for $\lambda = 1.1$ which clearly prooves that we have the exact solution.

Table 4.7. Optimal Results with cut method for 2A0B and $\rho = 0.1$

λ	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
Naïve									
0.1	1000000	0.09071	0.08249	0.00056	0.09015	0.09128	**C**		
0.1	100000	0.09042	0.08248	0.00178	0.08864	0.09220	**ARENA**		
IS									
0.1	1000000	0.09057	0.03930	0.00039	0.09019	0.09096	0.09057	0.0000	0.99
0.2	1000000	0.09054	0.01670	0.00025	0.09028	0.09079	0.09054	0.0000	0.50
0.3	1000000	0.09060	0.00926	0.00019	0.09041	0.09079	0.09060	0.0000	0.33
0.4	1000000	0.09069	0.00561	0.00015	0.09054	0.09083	0.09069	0.0000	0.25
0.5	1000000	0.09076	0.00349	0.00012	0.09064	0.09088	0.09076	0.0000	0.20
0.6	1000000	0.09080	0.00215	0.00009	0.09071	0.09089	0.09080	0.0000	0.16
0.7	1000000	0.09084	0.00126	0.00007	0.09077	0.09091	0.09084	0.0000	0.14
0.8	1000000	0.09088	0.00066	0.00005	0.09083	0.09093	0.09088	0.0004	0.12
0.9	1000000	0.09088	0.00028	0.00003	0.09085	0.09092	0.09088	0.0039	0.11
1.0	1000000	0.09093	0.00006	0.00002	0.09091	0.09094	0.09093	0.0222	0.10
1.1	1000000	0.09045	**0.00000**	**0.00000**	0.09145	0.09045	0.09045	0.09091	0.09
1.2	1000000	0.09097	0.00006	0.00001	0.09096	0.09099	0.09097	0.0833	0.29
1.3	1000000	0.09089	0.00028	0.00003	0.09085	0.09092	0.09089	0.0769	0.77
1.4	1000000	0.09087	0.00067	0.00005	0.09082	0.09092	0.09087	0.0714	1.78
1.5	1000000	0.09096	0.00123	0.00007	0.09089	0.09102	0.09096	0.0666	3.66
1.6	1000000	0.09085	0.00210	0.00009	0.09076	0.09094	0.09085	0.0625	6.84
1.7	1000000	0.09091	0.00334	0.00011	0.09079	0.09102	0.09091	0.0588	11.82
1.8	1000000	0.09081	0.00519	0.00014	0.09067	0.09095	0.09081	0.0555	19.16
1.9	1000000	0.09085	0.00798	0.00018	0.09068	0.09103	0.09085	0.0526	29.43

Because of the dependence we have most of the maximum weights smaller than one. Moreover, the average weights are exactly the same with estimator $\hat{\theta}$.

We show the variance graph of cut method for different φ values in Figure 4.3.

We have very close results for different φ values. To compare the results of cut method with our previous results we give Figure 4.4. In Figure 4.4, we put the graphs of naïve simulation results and the results of dependent Exponential IS density which is the previous best IS density.

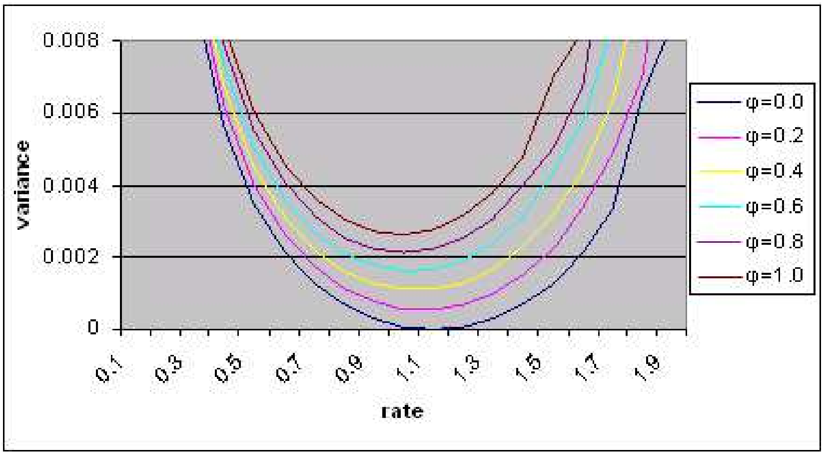


Figure 4.3. Variance graph for cut method with different φ values for 2A0B

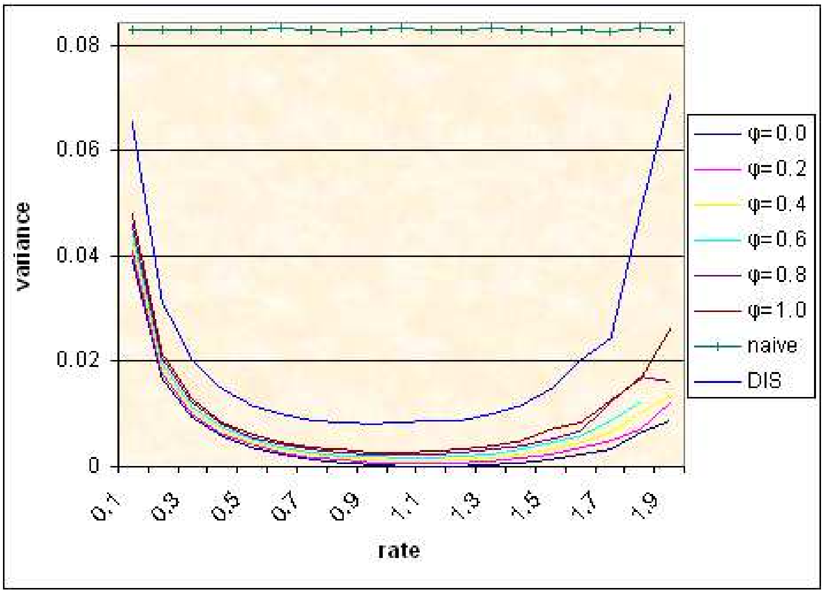


Figure 4.4. Variance graph for cut method with naive result

Although dependent Exponential IS density gives good results, the cut method outperforms clearly.

4.2.2. Results for Some Higher Dimensions

We give the results for three arrivals and buffer size of one case and $\varphi = 0$ in Table 4.8 that we have again very good results. For $\lambda = 1.1$ we have a zero variance estimator. We have a three dimensional integral for this case. The estimator has zero variance because of the dependence for even higher values of arrival rates.

Table 4.8. Results with cut method for 3A1B and $\rho = 0.1$

λ	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
Naïve									
0.1	1000000	0.00820	0.00813	0.00018	0.00802	0.00837	**C**		
0.1	100000	0.00809	0.00793	0.00055	0.00754	0.00864	**ARENA**		
IS									
0.1	1000000	0.008275	0.002207	0.000092	0.008183	0.00836	0.00827	0.0000	0.986
0.2	1000000	0.008263	0.000555	0.000046	0.008217	0.00830	0.00826	0.0000	0.249
0.3	1000000	0.008255	0.000239	0.000030	0.008225	0.00828	0.00825	0.0000	0.111
0.4	1000000	0.008275	0.000125	0.000022	0.008253	0.00829	0.00827	0.0000	0.062
0.5	1000000	0.008257	0.000070	0.000016	0.008241	0.00827	0.00825	0.0000	0.039
0.6	1000000	0.008263	0.000040	0.000012	0.008250	0.00827	0.00826	0.0000	0.027
0.7	1000000	0.008267	0.000022	0.000009	0.008257	0.00827	0.00826	0.0000	0.020
0.8	1000000	0.008259	0.000011	0.000007	0.008252	0.00826	0.00825	0.0002	0.015
0.9	1000000	0.008266	0.000005	0.000004	0.008262	0.00827	0.00826	0.0002	0.012
1.0	1000000	0.008265	0.000001	0.000002	0.008263	0.00826	0.00826	0.0020	0.010
1.1	1000000	0.008289	**0.000000**	**0.000000**	0.00828	0.00828	0.008289	0.0082	0.008
1.2	1000000	0.008267	0.000001	0.000002	0.008265	0.00826	0.00826	0.0069	0.030
1.3	1000000	0.008265	0.000005	0.000004	0.008260	0.00826	0.00826	0.0059	0.104
1.4	1000000	0.008263	0.000011	0.000007	0.008256	0.00826	0.00826	0.0051	0.173
1.5	1000000	0.008274	0.000022	0.000009	0.008264	0.00828	0.00827	0.0044	0.494
1.6	1000000	0.008262	0.000040	0.000012	0.008249	0.00827	0.00826	0.0039	1.274
1.7	1000000	0.008256	0.000064	0.000016	0.008240	0.00827	0.00825	0.0034	0.978
1.8	1000000	0.008273	0.000130	0.000022	0.008251	0.00829	0.00827	0.0030	4.706
1.9	1000000	0.008279	0.000195	0.000027	0.008252	0.00830	0.00827	0.0027	4.720

Although the cut method provides a run time advantage, it is applicable for only some special cases. Cut Method works for (Buffer Size=Number of Arrivals – 2). We use the sum of the interarrival times to generate dependent service times. The random number based approach may cause to have interarrival times in a large scale and consequently the dependent service times can be affected from this. Then the estimator is not stable. We show the results for some higher dimensions d (Buffer Size $= d - 2$) in Table 4.9. Up to dimension 10 we have sensible results. For higher dimensions the cut method has numerical problems. We all have the results for $\varphi = 0$ in Table 4.9.

As a future work, dependent algorithms with sequential sampling from the original density can be used for higher dimensional problems, [18]. Anyway it is not easy to have very large factor of variance reduction for higher dimensional problems.

Table 4.9. Results with cut method for some high dimensions

	$\hat{\theta}$	MSE	HL	LB	UB
Naïve d=4, ρ=0.1	0.00074500	0.00074445	0.00005348	0.00069152	0.00079848
Cut Method d=4, λ^*=1.1	0.00074820	0.00000000	0.00000011	0.00074810	0.00074831
Naïve d=5, ρ=0.2	0.00077000	0.00076941	0.00005437	0.00071563	0.00082437
Cut Method d=5, λ^*=1.2	0.00077697	0.00000000	0.00000000	0.00077697	0.00077697
Naïve d=6, ρ=0.2	0.00012597	0.00013698	0.00002294	0.00010303	0.00014891
Cut Method d=6, λ^*=1.2	0.00012926	0.00000000	0.00000000	0.00012926	0.00012926
Naïve d=7, ρ=0.7	0.00494100	0.00491659	0.00013743	0.00480357	0.00507843
Cut Method d=7,λ^*=1.7	0.00488220	0.00000000	0.00000000	0.00488220	0.00488220
Naïve d=8, ρ=0.7	0.00204037	0.00187746	0.00008493	0.00195545	0.00212530
Cut Method d=8, λ^*=1.7	0.00200715	0.00000010	0.00000062	0.00200654	0.00200777
Naïve d=9, ρ=0.7	0.00082311	0.00029497	0.00003366	0.00078945	0.00085677
Cut Method d=9, λ^*=1.7	0.00083577	0.00000000	0.00000000	0.00083577	0.00083577
Naïve d=10, ρ=0.7	0.00032696	0.00010220	0.00001981	0.00030715	0.00034678
Cut Method d=10, λ^*=1.7	0.00033788	0.00000000	0.00000011	0.00033777	0.00033799

5. IMPLEMENTATION OF IMPORTANCE SAMPLING WITH C AND ARENA

We implement IS with C and ARENA for the M/M/1 queueing system. As we do not know any ARENA implementation of IS we give the details below.

5.1. ARENA Implementation

ARENA enables an easy modelling capability. Although it seemed very difficult to implement IS with ARENA at the very beginning of the thesis, the resulting model is so simple. We give the whole model in Figure 5.1. We had hundredthousand replications for ARENA. We create customers in the *create module* having previously specified interarrival time distributions. We have one *process module* including a limited resource queue. And finally we have a *dispose module* that served or quitted customers are disposed.

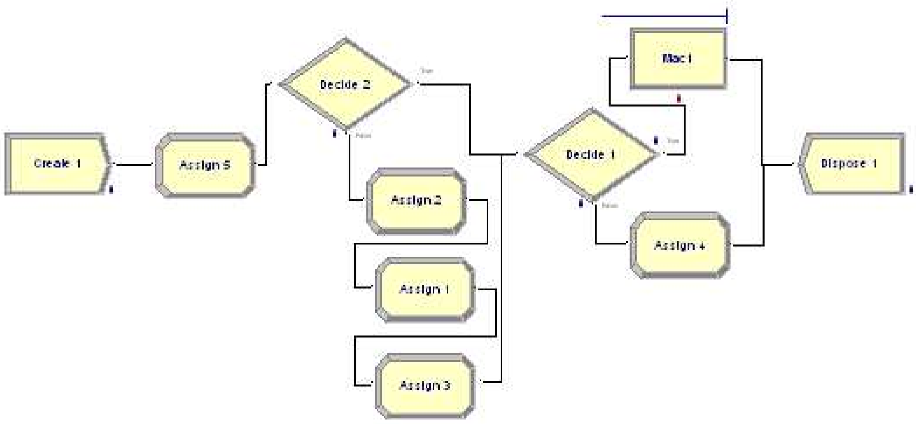


Figure 5.1. Whole Model - IS Implementation with ARENA

We have two *decide modules* that one identifies the first customer and send it directly to the server. The other one is used for checking the buffer. If the buffer is full then the customer is disposed before service. The implementation of IS is done with the *assign modules* in the model. Assign modules in the model are called according to their priority such as Assign1, Assign2 and so on. We include all assign modules in accordance to understand the model better. Indeed there is no need to have seperate assign modules unless we have another module between the two assign modules. We have a variable *PrewTime* in Assign1 to hold the simulation time of the prewious customer as shown in Figure 5.2. TNOW is an ARENA variable that holds the simulation time. Default initial values of all variables are zero in ARENA.

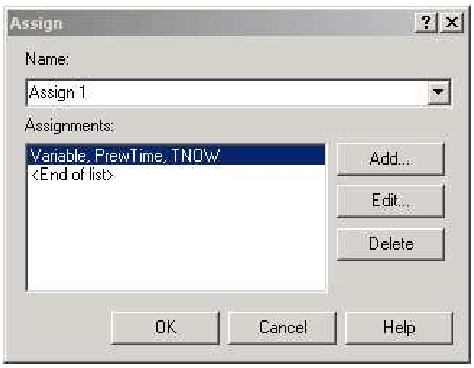


Figure 5.2. Holding previous customer's passing time

We hold the interarrival times with the variable *interarrivaltime* as substracting the prewious customer's simulation time from the current simulation time as shown in Figure 5.3.

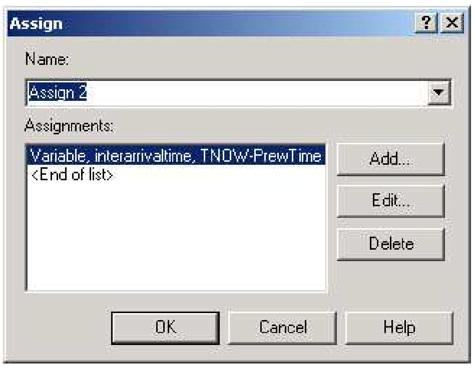


Figure 5.3. Holding interarrival times as a variable

We have an attribute *weight* that holds the likelihood ratio of $\frac{f(x)}{g(x)}$ for the IS as shown in Figure 5.4. The reason we use an attribute is that attribute uses instantenous value of the variable whereas a variable uses the final value in statistics while calculating any expression.

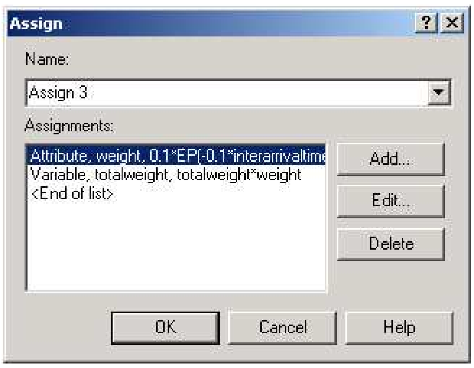


Figure 5.4. Weight calculation

When we double click on the weight attribute in Figure 5.4 we have the expression of the weight function in IS. Weight expression can be seen in Figure 5.5 below.

Figure 5.5. Weight and Likelihood Ratio

And we check any bufer overflow with variable qxx in assign 4 as shown in Figure 5.6. The variable q can be used for estimating the number of quitted customers in the system.

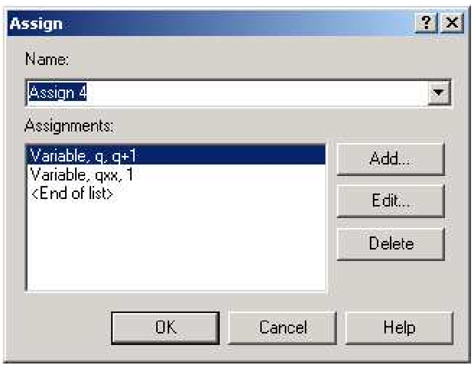
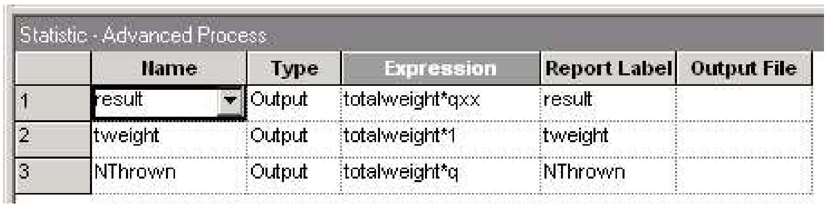


Figure 5.6. Checking buffer overflow

ARENA enables a powerful statistics module for the statistics calculations. We have

$$\hat{\theta} = totalweight * qxx \tag{5.1}$$

for the result of the simulation. Statistics module can be seen in Figure 5.7.


Statistic - Advanced Process

	Name	Type	Expression	Report Label	Output File
1	result	Output	totalweight*qxx	result	
2	tweight	Output	totalweight*1	tweight	
3	NThrown	Output	totalweight*q	NThrown	

Figure 5.7. Statistics calculation

5.2. C Implementation

We have written an efficient simulation program for simple M/M/1 queue. As C is one of the best programing language with superior run time advantage, we could have one million replications. We changed the seed for generating safely random numbers for different parameters. We hold a pointer to determine any buffer overflow during the simulation. We put the code of Independent Exponential IS density and Uniform random number generator files of Law and Kelton [16] into Appendix D and all other codes for different IS densities to the Electronic Appendix. We have an input txt file for the parameters of the IS densities and for some statistics calculations of the system. And we have an output txt file displaying the estimator, MSE, average weights, minimum weights and maximum weights.

6. CONCLUSIONS

Simulation is simple for solving the complex problems but variance reduction is very difficult for those problems. We used Importance Sampling (IS) technique for queueing systems. Independent IS (IIS) is simple but variance reduction is moderate. Dependent IS (DIS) is more difficult, variance reduction is larger but it is still not very large for high dimensions. One contribution of this thesis is that we use defensive mixture of dependent and independent IS densities which leads to very good results. We use the original distribution as one componenet of the IS distribution.

This thesis introduces a new dependent sampling algorithm for calculating some expectations of interest with Importance Sampling Method. The new approach is called Cut Method as we cut the sampling region in order to identify the important part which has a dominant effect on the value of the expectation. But it can be used for only some special cases of (Buffer Size=Number of Arrivals -2). Up to dimension ten the new approach works well, and for higher dimensions it starts to give violated results. Then dependent algorithms with sequential sampling from the original density can be used for higher dimensional problems.

We used IS for both interarrival times and service times. Interarrival times IS should be also useful for larger queueing networks. IS can lead to variance reduction for queueing simulations. As a summary, whole optimal results that we had during thesis studies are given in Table 6.1 for IIS and in Table 6.2 for DIS.

As it is seen from the Tables, we have higher factor of variance reduction with Dependent IS. Dependent IS gives better results than Independent IS, but the combination of these two is superior especially for higher dimensions.

Another contribution of this thesis is that we implement the IS method with ARENA software for the literature. Experiments show that there is good agreement between theory and practise. Importance Sampling makes rare events less rare in order

Table 6.1. All Optimal Results for Independent IS

	$\hat{\theta}$	MSE	LB	UB	Factor
2A0B d=2 $\rho = 0.1$ Naïve C, λ=0.1	0.09071	0.08249	0.09015	0.09128	
Naïve ARENA, λ=0.1	0.09042	0.08248	0.08864	0.09220	
IS Exponential ($\lambda = 0.62$)	0.09067	0.01919	0.09040	0.09094	4.3
IS Pareto($\alpha = 18, \beta = 2.5$)	0.09032	0.07092	0.08980	0.09084	1.16
IS Def-Mix Exp&Uniform ($b = 3.2$), p=0.3	0.09067	0.03310	0.09032	0.09103	2.5
IS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 1.7$) p=0.0	0.09104	0.01973	0.09077	0.09132	4.27
3A1B d=3 $\rho = 0.1$ Naïve C, λ=0.1	0.00820	0.00813	0.00802	0.00837	
Naïve ARENA, λ=0.1	0.00809	0.00793	0.00754	0.00864	
IS Exponential ($\lambda = 0.67$)	0.00825	0.00066	0.00820	0.00830	12.31
IS Pareto($\alpha = 16.5, \beta = 3$)	0.00817	0.00513	0.00803	0.00831	1.59
IS Def-Mix Exp&Uniform ($b = 1.7$), p=0.4	0.00828	0.00174	0.00820	0.00836	4.68
IS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 2$) p=0.0	0.00830	0.00073	0.00825	0.00836	11.14
11A5B d=15 $\rho = 0.7$ Naïve C, λ=0.7	0.05143	0.04879	0.05100	0.05186	
Naïve ARENA, λ=0.7	0.05193	0.04957	0.05055	0.05331	
IS Exponential ($\lambda = 0.87$)	0.05131	0.03020	0.05097	0.05166	1.62
IS Pareto($\alpha = 19.5, \beta = 17.5$)	0.05089	0.03020	0.05055	0.05123	1.62
IS Def-Mix Exp&Uniform ($b = 1.0$), p=0.8	0.05125	0.03705	0.05087	0.05163	1.32
IS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 0.9$) p=0.6	0.05142	0.02958	0.05109	0.05176	1.65
101A20B d=180 $\rho = 0.7$ Naïve C, λ=0.7	0.00265	0.00264	0.00255	0.00275	
Naïve ARENA, λ=0.7	0.00263	0.00234	0.00233	0.00293	
IS Exponential($\lambda = 0.76$)	0.00260	0.00127	0.00253	0.00267	2.09
IS Pareto($\alpha = 28.5, \beta = 21.5$)	0.00260	0.00126	0.00253	0.00267	2.09
IS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 0.9$) p=0.8	0.00262	0.00124	0.00255	0.00269	2.13
101A22B d=178 $\rho = 0.7$ Naïve C, λ=0.7	0.001140	0.001139	0.001074	0.001206	
Naïve ARENA, λ=0.7	0.001180	0.001179	0.000879	0.001481	
IS Exponential($\lambda = 0.8$)	0.001135	0.000468	0.001093	0.001178	2.43
IS Pareto($\alpha = 29,\ \beta = 22$)	0.001117	0.000431	0.001077	0.001158	2.64
IS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 1.2$) p=0.6	0.001159	0.000502	0.002525	0.002622	2.27

to sample more from important regions when using the same sample size. Hence, we have a lower error for predicting the expectation of the rare event. As IS performs well

Table 6.2. All Optimal Results for Dependent IS

	$\hat{\theta}$	MSE	LB	UB	Factor
2A0B d=2 $\rho = 0.1$					
Naïve C, λ=0.1	0.09071	0.08249	0.09015	0.09128	
Naïve ARENA, λ=0.1	0.09042	0.08248	0.08864	0.09220	
DIS Exponential($\lambda = 0.9$),c=0.7	0.09087	0.00810	0.09069	0.09105	10.18
DIS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 1.2$)	0.09093	0.00915	0.09074	0.09112	9.02
IS Def-Mix Dept&Indept Exp($\lambda = 0.9$)	0.09094	0.00813	0.09076	0.09112	10.14
p=0,c=0.7					
3A1B d=3 $\rho = 0.1$					
Naïve C, λ=0.1	0.008198	0.008131	0.008021	0.008375	
Naïve ARENA, λ=0.1	0.00809	0.00793	0.00754	0.00864	
DIS Exponential($\lambda = 0.8$),c=0.5	0.008235	0.000316	0.0082	0.00827	25.75
DIS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 1.4$)	0.008293	0.000282	0.00826	0.008326	28.84
IS Def-Mix Dept&Indept Exp($\lambda = 0.8$)	0.008264	0.000324	0.008229	0.0083	25.08
p=0,c=0.5					
11A5B d=15 $\rho = 0.7$					
Naïve C, λ=0.7	0.05143	0.04879	0.05100	0.05186	
Naïve ARENA, λ=0.7	0.05193	0.04957	0.05055	0.05331	
DIS Exponential($\lambda = 0.88$),c=0.09	0.05146	0.02693	0.05114	0.05178	1.81
DIS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 1.3$)	0.05146	0.01984	0.05119	0.05174	2.50
IS Def-Mix Dept&Indept Exp($\lambda = 1.2$)	0.05143	0.02445	0.05112	0.05173	2.00
p=0.4,c=0.09					
101A20B d=180 $\rho = 0.7$					
Naïve C, λ=0.7	0.00265	0.00264	0.00255	0.00275	
Naïve ARENA, λ=0.7	0.00263	0.00234	0.00233	0.00293	
DIS Exponential($\lambda = 0.8$),c=0.03	0.00266	0.00129	0.00259	0.00273	2.05
DIS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 1.3$)	0.00257	0.00062	0.00252	0.00262	4.28
IS Def-Mix Dept&Indept Exp($\lambda = 1.0$)	0.00256	0.00085	0.00250	0.00262	2.74
p=0.6,c=0.09					
101A22B d=178 $\rho = 0.7$					
Naïve C, λ=0.7	0.001140	0.001139	0.001074	0.001206	
Naïve ARENA, λ=0.7	0.001210	0.001049	0.000974	0.001246	
DIS Exponential($\lambda = 0.8$)	0.001164	0.000405	0.001125	0.001204	2.81
DIS Def-Mix Exp&Gamma ($\alpha = 0.9, \beta = 1.1$)	0.001116	0.000315	0.002525	0.002622	3.61
IS Def-Mix Dept&Indept Exp($\lambda = 0.9$)	0.001136	0.000301	0.001081	0.001151	3.78
p=0.5					

for exceedance probabilities of rare events which occur at the tails of the distributions like Exponential, heavy tailed IS densities provide good variance reductions.

APPENDIX A: VARIANCE FORMULA FOR IMPORTANCE SAMPLING

As it is mentioned in section (2.1) the general idea of IS is:

$$E_f\left[q\left(X\right)\right] = E_g\left[q\left(X\right)\frac{f\left(x\right)}{g\left(x\right)}\right] \tag{A.1}$$

If we look at the second moment of the right hand side of equation A.1, we have the following equation:

$$E_g\left[\left(q\left(X\right)\frac{f\left(x\right)}{g\left(x\right)}\right)^2\right] = E_f\left[\left(q\left(X\right)\right)^2\frac{f\left(x\right)}{g\left(x\right)}\right] \tag{A.2}$$

Then the general variance formula can be written using the original density as:

$$\begin{aligned} var_f\left(x\right) &= E_f\left[x^2\right] - \left(E_f\left[x\right]\right)^2 \\ &= \int x^2 f\left(x\right)dx - \theta^2 \\ &= \int \left(q\left(X\right)^2\right)\frac{f\left(x\right)}{g\left(x\right)}f\left(x\right)dx - \theta^2 \end{aligned} \tag{A.3}$$

We have the same formula using the IS density:

$$\begin{aligned} var_g\left(x\right) &= E_g\left[x^2\right] - \left(E_g\left[x\right]\right)^2 \\ &= \int x^2 g\left(x\right)dx - \theta^2 \\ &= \int \left(q\left(X\right)\frac{f\left(x\right)}{g\left(x\right)}\right)^2 g\left(x\right)dx - \theta^2 \\ &= \int \left(q\left(X\right)^2\right)\frac{f\left(x\right)}{g\left(x\right)}f\left(x\right)dx - \theta^2 \end{aligned} \tag{A.4}$$

APPENDIX B: RESULTS FOR CHAPTER 3

Table B.1. Close to Optimal Results for Exponential IS density for 2A0B and $\rho = 0.1$

λ	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
0.50	1000000	0.09083	0.02026	0.00028	0.09055	0.09111	0.96875	0.20000	33237.6
0.51	1000000	0.09079	0.02007	0.00028	0.09051	0.09107	0.96662	0.19608	34563.8
0.52	1000000	0.09085	0.01990	0.00028	0.09057	0.09112	0.95701	0.19231	35875.2
0.53	1000000	0.09087	0.01988	0.00028	0.09060	0.09115	0.97076	0.18868	37170.7
0.54	1000000	0.09089	0.01978	0.00028	0.09062	0.09117	0.96594	0.18519	38448.9
0.55	1000000	0.09088	0.01964	0.00027	0.09060	0.09115	1.00528	0.18182	39708.8
0.56	1000000	0.09086	0.01951	0.00027	0.09059	0.09114	1.00494	0.17857	40949.6
0.57	1000000	0.09097	0.01948	0.00027	0.09069	0.09124	1.00031	0.17544	42170.3
0.58	1000000	0.09093	0.01943	0.00027	0.09066	0.09121	0.98482	0.17241	43370.3
0.59	1000000	0.09097	0.01948	0.00027	0.09069	0.09124	0.98173	0.16949	44549.1
0.60	1000000	0.09103	0.01947	0.00027	0.09075	0.09130	1.01404	0.16667	45706.0
0.61	1000000	0.09095	0.01940	0.00027	0.09067	0.09122	1.01465	0.16394	46840.9
0.62	1000000	0.09089	**0.01929**	0.00027	0.09062	0.09117	1.01066	0.16129	47953.2
0.63	1000000	0.09091	0.01944	0.00027	0.09064	0.09119	1.00720	0.15873	49042.9
0.64	1000000	0.09097	0.01944	0.00027	0.09070	0.09124	1.00985	0.15625	50109.6
0.65	1000000	0.09097	0.01958	0.00027	0.09070	0.09125	1.01427	0.15385	51153.4
0.66	1000000	0.09102	0.01980	0.00028	0.09075	0.09130	1.01329	0.15152	52174.2
0.67	1000000	0.09107	0.01989	0.00028	0.09080	0.09135	1.01034	0.14925	53171.9
0.68	1000000	0.09115	0.02008	0.00028	0.09088	0.09143	1.01142	0.14706	54146.7
0.69	1000000	0.09121	0.02034	0.00028	0.09093	0.09149	1.00471	0.14493	55098.5
0.70	1000000	0.09118	0.02049	0.00028	0.09090	0.09146	0.95059	0.14286	56027.6

Table B.2. Results for Pareto IS density with $\beta = 2.5$ for 2A0B and $\rho = 0.1$

α	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
12.0	1000000	0.09122	0.07908	0.00055	0.09067	0.09177	1.00045	0.00	2.083
12.5	1000000	0.09075	0.07738	0.00055	0.09021	0.09130	1.00087	0.00	2.000
13.0	1000000	0.09108	0.07619	0.00054	0.09054	0.09162	0.99872	0.00	1.923
13.5	1000000	0.09138	0.07569	0.00054	0.09084	0.09192	1.00067	0.00	1.852
14.0	1000000	0.09083	0.07400	0.00053	0.09030	0.09137	0.99876	0.00	1.786
14.5	1000000	0.09084	0.07353	0.00053	0.09031	0.09137	0.99986	0.00	1.724
15.0	1000000	0.09078	0.07255	0.00053	0.09025	0.09130	1.00033	0.00	1.667
15.5	1000000	0.09094	0.07231	0.00053	0.09041	0.09146	1.00030	0.00	1.613
16.0	1000000	0.09096	0.07205	0.00053	0.09044	0.09149	0.99953	0.00	1.563
16.5	1000000	0.09134	0.07184	0.00053	0.09082	0.09187	1.00006	0.00	1.515
17.0	1000000	0.09113	0.07155	0.00052	0.09060	0.09165	0.99975	0.00	1.471
17.5	1000000	0.09062	0.07103	0.00052	0.09010	0.09114	0.99996	0.00	1.429
18.0	1000000	0.09032	**0.07092**	0.00052	0.08980	0.09084	1.00028	0.00	1.389
18.5	1000000	0.09122	0.07178	0.00053	0.09070	0.09175	0.99971	0.00	1.351
19.0	1000000	0.09119	0.07184	0.00053	0.09067	0.09172	0.99994	0.00	1.316
19.5	1000000	0.09088	0.07197	0.00053	0.09035	0.09140	0.99965	0.00	1.282
20.0	1000000	0.09117	0.07258	0.00053	0.09064	0.09170	1.00008	0.00	1.250

.

.

Table B.3. Results for defensive mixture of Exponential and Gamma IS densities with $p = 0.0$ and $\alpha = 0.9$ for 2A0B, $\rho = 0.1$

β	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
1.2	1000000	0.09098	0.02558	0.00031	0.09067	0.09130	0.82437	0.02722	32664.2
1.3	1000000	0.09106	0.02526	0.00031	0.09075	0.09137	0.88158	0.02553	23470.9
1.4	1000000	0.09093	0.02221	0.00029	0.09064	0.09122	0.91383	0.02433	34381.6
1.5	1000000	0.09081	0.02130	0.00029	0.09053	0.09110	0.92312	0.02946	52971.9
1.6	1000000	0.09089	0.02060	0.00028	0.09061	0.09117	0.91551	0.03452	25030.9
1.7	1000000	0.09078	0.01930	0.00027	0.09051	0.09106	0.96194	0.04261	17860.3
1.8	1000000	0.09089	0.01986	0.00028	0.09061	0.09117	0.99307	0.03741	40978.2
1.9	1000000	0.09104	**0.01973**	0.00028	0.09077	0.09132	0.91283	0.04099	17116.5
2.0	1000000	0.09094	0.01977	0.00028	0.09066	0.09121	0.97815	0.03927	21144.5
2.1	1000000	0.09099	0.02049	0.00028	0.09071	0.09127	0.92283	0.04365	3799.6
2.2	1000000	0.09084	0.02022	0.00028	0.09056	0.09112	0.94618	0.03824	6989.4
2.3	1000000	0.09081	0.02064	0.00028	0.09053	0.09109	0.98647	0.04780	25189.5
2.4	1000000	0.09105	0.02105	0.00028	0.09076	0.09133	0.98698	0.04713	14466.3
2.5	1000000	0.09095	0.02143	0.00029	0.09066	0.09123	1.00089	0.05547	30074.1
2.6	1000000	0.09070	0.02180	0.00029	0.09041	0.09099	1.01168	0.05898	26881.3
2.7	1000000	0.09090	0.02229	0.00029	0.09061	0.09120	0.99067	0.06649	5027.1
2.8	1000000	0.09101	0.02273	0.00030	0.09071	0.09131	0.98011	0.04867	8315.9
2.9	1000000	0.09053	0.02313	0.00030	0.09024	0.09083	1.00714	0.05040	13199.1
3.0	1000000	0.09119	0.02383	0.00030	0.09088	0.09149	0.98422	0.07520	2057.6

Table B.4. Results for defensive mixture of Exponential and Uniform IS densities for $p = 0.3$, 2A0B and $\rho = 0.1$

b	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
2.3	1000000	0.09091	0.03950	0.00039	0.09052	0.09129	0.99807	0.24210	3.33333
2.4	1000000	0.09091	0.03798	0.00038	0.09053	0.09129	0.99792	0.24951	3.33333
2.5	1000000	0.09108	0.03752	0.00038	0.09070	0.09146	0.99889	0.25672	3.33333
2.6	1000000	0.09108	0.03638	0.00037	0.09071	0.09145	0.99858	0.26373	3.33333
2.7	1000000	0.09097	0.03561	0.00037	0.09060	0.09134	0.99838	0.27055	3.33333
2.8	1000000	0.09090	0.03479	0.00037	0.09053	0.09126	0.99985	0.27718	3.33333
2.9	1000000	0.09070	0.03379	0.00036	0.09034	0.09106	0.99945	0.28362	3.33333
3.0	1000000	0.09047	0.03339	0.00036	0.09011	0.09083	0.99920	0.28988	3.33333
3.1	1000000	0.09064	0.03318	0.00036	0.09029	0.09100	0.99720	0.29597	3.33333
3.2	1000000	0.09067	**0.03310**	0.00036	0.09032	0.09103	0.99573	0.30189	3.33333
3.3	1000000	0.09070	0.03337	0.00036	0.09034	0.09106	0.99586	0.30764	3.33333
3.4	1000000	0.09080	0.03361	0.00036	0.09044	0.09116	0.99737	0.31323	3.33333
3.5	1000000	0.09081	0.03393	0.00036	0.09045	0.09117	0.99742	0.31866	3.33333
3.6	1000000	0.09089	0.03427	0.00036	0.09053	0.09125	0.99719	0.32394	3.33333
3.7	1000000	0.09074	0.03467	0.00036	0.09038	0.09111	0.99742	0.32906	3.33333
3.8	1000000	0.09084	0.03507	0.00037	0.09048	0.09121	0.99679	0.33404	3.33333
3.9	1000000	0.09089	0.03561	0.00037	0.09052	0.09126	0.99785	0.33887	3.33333
4.0	1000000	0.09079	0.03596	0.00037	0.09042	0.09116	0.99883	0.34356	3.33333
4.1	1000000	0.09088	0.03665	0.00038	0.09051	0.09126	0.99860	0.34812	3.33333

Table B.5. Results for defensive mixture of Exponential and Gamma IS density for 2A0B, p=0, and $\rho = 0.1$

β	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
α=0.5									
0.8	1000000	0.09053	0.15225	0.00076	0.08976	0.09129	0.61141	0.0	22125.5
0.9	1000000	0.09046	0.09165	0.00059	0.08987	0.09106	0.82043	0.0	96246.6
1.0	1000000	0.09183	0.28264	0.00104	0.09079	0.09287	0.68756	0.0	28003.4
1.1	1000000	0.09066	0.05629	0.00047	0.09020	0.09113	0.72163	0.0	22118.2
1.2	1000000	0.09076	0.06316	0.00049	0.09026	0.09125	1.00492	0.0	207288.2
1.3	1000000	0.09127	0.05343	0.00045	0.09081	0.09172	0.74833	0.0	13134.9
1.4	1000000	0.09129	0.04341	0.00041	0.09088	0.09170	0.79487	0.0	22605.3
1.5	1000000	0.09082	0.03784	0.00038	0.09044	0.09120	0.90722	0.0	84226.1

Continued on Next Page…

β	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
1.6	1000000	0.09093	0.03753	0.00038	0.09055	0.09131	0.85833	0.0	41113.7
1.7	1000000	0.09079	0.03273	0.00035	0.09043	0.09114	1.01275	0.0	108085.1
1.8	1000000	0.09089	0.03184	0.00035	0.09054	0.09124	0.87909	0.0	18733.6
1.9	1000000	0.09088	0.03094	0.00034	0.09053	0.09122	0.92090	0.0	48034.1
2.0	1000000	0.09098	0.02912	0.00033	0.09065	0.09131	0.91115	0.0	19069.5
2.1	1000000	0.09079	0.02825	0.00033	0.09046	0.09112	0.90194	0.0	9485.2
2.2	1000000	0.09085	0.02747	0.00032	0.09052	0.09117	0.98987	0.0	28532.9
2.3	1000000	0.09079	0.02689	0.00032	0.09047	0.09111	0.94911	0.0	20223.4
2.4	1000000	0.09114	0.02722	0.00032	0.09082	0.09147	0.97191	0.0	47449.0
2.5	1000000	0.09064	0.02640	0.00032	0.09032	0.09095	1.04301	0.0	102266.4
2.6	1000000	0.09080	0.02607	0.00032	0.09048	0.09111	1.02392	0.0	91840.4
2.7	1000000	0.09091	0.02607	0.00032	0.09060	0.09123	1.08764	0.0	82355.4
2.8	1000000	0.09106	0.02583	0.00032	0.09075	0.09138	0.95967	0.0	24343.4
2.9	1000000	0.09076	0.02577	0.00031	0.09045	0.09108	0.97220	0.0	11521.6
3.0	1000000	0.09072	0.02524	0.00031	0.09041	0.09103	0.97000	0.0	7078.8
3.1	1000000	0.09138	0.02564	0.00031	0.09106	0.09169	0.98057	0.0	4763.1
3.2	1000000	0.09075	0.02531	0.00031	0.09044	0.09106	1.03047	0.0	32528.6
3.3	1000000	0.09082	0.02525	0.00031	0.09050	0.09113	1.01827	0.0	10552.2
3.4	1000000	0.09105	0.02538	0.00031	0.09074	0.09136	0.99572	0.0	6205.3
3.5	1000000	0.09120	0.02558	0.00031	0.09089	0.09152	0.99361	0.0	8148.8
3.6	1000000	0.09092	0.02527	0.00031	0.09060	0.09123	0.98218	0.0	4541.2
3.7	1000000	0.09072	0.02531	0.00031	0.09041	0.09103	1.01551	0.0	18524.6
3.8	1000000	0.09114	0.02550	0.00031	0.09083	0.09145	1.00032	0.0	6868.4
3.9	1000000	0.09101	0.02559	0.00031	0.09069	0.09132	0.99469	0.0	5188.2
4.0	1000000	0.09129	0.02577	0.00031	0.09098	0.09161	0.99104	0.0	1965.1
4.1	1000000	0.09076	0.02576	0.00031	0.09044	0.09107	1.02132	0.0	31771.5
α=0.6									
0.8	1000000	0.09049	0.11544	0.00067	0.08983	0.09116	1.45460	0.00000	831541.8
0.9	1000000	0.09045	0.07671	0.00054	0.08990	0.09099	0.72371	0.00001	29989.0
1.0	1000000	0.09065	0.06184	0.00049	0.09016	0.09113	0.81278	0.00002	61320.8
1.1	1000000	0.09097	0.05386	0.00045	0.09052	0.09143	0.71156	0.00001	16720.9
1.2	1000000	0.09075	0.04452	0.00041	0.09034	0.09117	0.74929	0.00001	14830.5
1.3	1000000	0.09069	0.03815	0.00038	0.09031	0.09108	1.48591	0.00003	609431.6
1.4	1000000	0.09064	0.03267	0.00035	0.09029	0.09100	0.80246	0.00002	8209.1
1.5	1000000	0.09095	0.03184	0.00035	0.09060	0.09130	0.97136	0.00002	124771.4

Continued on Next Page. . .

β	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
1.6	1000000	0.09094	0.02888	0.00033	0.09061	0.09128	0.86526	0.00002	17039.7
1.7	1000000	0.09108	0.02764	0.00033	0.09076	0.09141	0.96108	0.00001	46741.2
1.8	1000000	0.09056	0.02528	0.00031	0.09025	0.09087	0.93361	0.00001	10991.2
1.9	1000000	0.09119	0.02583	0.00031	0.09087	0.09150	0.91256	0.00002	24755.4
2.0	1000000	0.09082	0.02419	0.00030	0.09052	0.09113	0.91635	0.00003	21394.2
2.1	1000000	0.09093	0.02423	0.00031	0.09063	0.09124	1.20415	0.00003	275359.8
2.2	1000000	0.09080	0.02349	0.00030	0.09050	0.09110	0.95978	0.00003	31014.0
2.3	1000000	0.09086	0.02346	0.00030	0.09056	0.09116	0.93130	0.00002	14920.1
2.4	1000000	0.09117	0.02399	0.00030	0.09086	0.09147	1.19116	0.00007	195166.3
2.5	1000000	0.09089	0.02317	0.00030	0.09059	0.09119	0.94891	0.00003	7644.4
2.6	1000000	0.09107	0.02287	0.00030	0.09077	0.09137	0.97083	0.00004	8536.4
2.7	1000000	0.09100	0.02287	0.00030	0.09070	0.09130	1.03051	0.00005	59340.9
2.8	1000000	0.09096	0.02290	0.00030	0.09067	0.09126	0.98051	0.00002	23326.9
2.9	1000000	0.09079	0.02273	0.00030	0.09049	0.09108	0.98781	0.00002	12115.8
3.0	1000000	0.09112	0.02300	0.00030	0.09083	0.09142	0.97547	0.00006	4471.9
3.1	1000000	0.09076	0.02294	0.00030	0.09046	0.09106	0.98757	0.00001	9990.0
3.2	1000000	0.09075	0.02314	0.00030	0.09046	0.09105	1.00002	0.00002	6114.9
3.3	1000000	0.09092	0.02348	0.00030	0.09062	0.09123	1.00475	0.00004	9680.2
3.4	1000000	0.09087	0.02334	0.00030	0.09057	0.09117	0.99859	0.00003	4087.7
3.5	1000000	0.09074	0.02342	0.00030	0.09044	0.09104	0.98864	0.00006	3618.5
3.6	1000000	0.09084	0.02369	0.00030	0.09053	0.09114	0.99948	0.00010	3874.6
3.7	1000000	0.09073	0.02374	0.00030	0.09043	0.09103	1.00360	0.00011	4822.9
3.8	1000000	0.09083	0.02399	0.00030	0.09052	0.09113	1.01835	0.00004	11744.5
3.9	1000000	0.09081	0.02407	0.00030	0.09051	0.09112	0.99051	0.00001	1724.7
4.0	1000000	0.09078	0.02432	0.00031	0.09047	0.09109	0.99082	0.00010	2261.1
4.1	1000000	0.09081	0.02449	0.00031	0.09051	0.09112	0.98691	0.00003	930.0
α=0.7									
0.8	1000000	0.09131	0.28641	0.00105	0.09026	0.09236	0.62295	0.00024	15248.3
0.9	1000000	0.09028	0.04716	0.00043	0.08985	0.09070	0.71763	0.00031	38209.9
1.0	1000000	0.09104	0.08332	0.00057	0.09047	0.09161	0.74000	0.00039	37240.7
1.1	1000000	0.09083	0.04199	0.00040	0.09043	0.09123	0.80322	0.00066	44585.5
1.2	1000000	0.09101	0.03557	0.00037	0.09064	0.09138	0.78149	0.00058	14605.7
1.3	1000000	0.09064	0.02896	0.00033	0.09030	0.09097	0.82317	0.00015	22396.0
1.4	1000000	0.09117	0.03982	0.00039	0.09078	0.09156	0.80580	0.00066	12063.7
1.5	1000000	0.09083	0.02588	0.00032	0.09052	0.09115	1.01344	0.00034	144906.2

Continued on Next Page…

β	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
1.6	1000000	0.09105	0.02541	0.00031	0.09074	0.09136	0.89612	0.00018	11567.3
1.7	1000000	0.09112	0.02358	0.00030	0.09082	0.09142	0.95431	0.00039	47821.8
1.8	1000000	0.09091	0.02246	0.00029	0.09062	0.09121	1.16881	0.00062	279222.4
1.9	1000000	0.09084	0.02186	0.00029	0.09055	0.09113	0.89915	0.00061	11945.6
2.0	1000000	0.09092	0.02151	0.00029	0.09063	0.09120	1.14430	0.00094	225061.6
2.1	1000000	0.09088	0.02161	0.00029	0.09059	0.09116	0.93004	0.00024	11807.2
2.2	1000000	0.09116	0.02145	0.00029	0.09088	0.09145	1.01100	0.00040	30959.7
2.3	1000000	0.09093	0.02113	0.00028	0.09065	0.09122	0.96290	0.00108	21060.4
2.4	1000000	0.09109	0.02138	0.00029	0.09080	0.09137	0.93061	0.00109	5870.0
2.5	1000000	0.09055	0.02114	0.00028	0.09026	0.09083	0.98834	0.00057	19887.0
2.6	1000000	0.09120	0.02155	0.00029	0.09091	0.09149	0.96157	0.00077	5281.2
2.7	1000000	0.09089	0.02148	0.00029	0.09060	0.09117	0.98532	0.00050	12944.7
2.8	1000000	0.09103	0.02166	0.00029	0.09074	0.09131	0.96913	0.00074	10704.8
2.9	1000000	0.09081	0.02182	0.00029	0.09052	0.09110	0.98085	0.00112	11823.8
3.0	1000000	0.09069	0.02203	0.00029	0.09040	0.09098	0.97926	0.00146	4829.0
3.1	1000000	0.09112	0.02230	0.00029	0.09083	0.09142	1.06449	0.00146	59689.8
3.2	1000000	0.09114	0.02251	0.00029	0.09084	0.09143	0.99629	0.00037	6668.5
3.3	1000000	0.09113	0.02275	0.00030	0.09083	0.09142	0.98648	0.00140	3316.8
3.4	1000000	0.09087	0.02299	0.00030	0.09058	0.09117	0.99148	0.00090	2852.3
3.5	1000000	0.09103	0.02327	0.00030	0.09073	0.09133	0.99936	0.00040	9630.3
3.6	1000000	0.09082	0.02343	0.00030	0.09052	0.09112	1.00152	0.00175	4441.4
3.7	1000000	0.09078	0.02376	0.00030	0.09048	0.09108	1.00000	0.00085	2502.6
3.8	1000000	0.09106	0.02410	0.00030	0.09075	0.09136	0.99095	0.00100	1296.3
3.9	1000000	0.09062	0.02416	0.00030	0.09032	0.09093	0.99879	0.00200	3247.4
4.0	1000000	0.09115	0.02465	0.00031	0.09084	0.09146	1.00242	0.00138	4651.0
4.1	1000000	0.09078	0.02485	0.00031	0.09047	0.09109	1.01134	0.00047	8119.9
α=0.8									
0.8	1000000	0.09130	0.17297	0.00082	0.09048	0.09211	0.71230	0.00269	32860.1
0.9	1000000	0.09097	0.06683	0.00051	0.09047	0.09148	0.74060	0.00175	21089.5
1.0	1000000	0.09073	0.04126	0.00040	0.09034	0.09113	0.81085	0.00416	40221.6
1.1	1000000	0.09095	0.04526	0.00042	0.09054	0.09137	0.88220	0.00389	67604.4
1.2	1000000	0.09079	0.03366	0.00036	0.09043	0.09115	0.83713	0.00442	26188.5
1.3	1000000	0.09059	0.02551	0.00031	0.09028	0.09090	1.28275	0.00497	335348.0
1.4	1000000	0.09078	0.02337	0.00030	0.09048	0.09108	0.87971	0.00552	13741.5
1.5	1000000	0.09100	0.02243	0.00029	0.09070	0.09129	1.07702	0.00383	108557.0

Continued on Next Page…

β	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
1.6	1000000	0.09068	0.02133	0.00029	0.09039	0.09097	0.93630	0.00505	32845.3
1.7	1000000	0.09080	0.02048	0.00028	0.09052	0.09108	0.91492	0.00738	21054.7
1.8	1000000	0.09087	0.02074	0.00028	0.09058	0.09115	0.98262	0.00688	28063.3
1.9	1000000	0.09103	0.02259	0.00029	0.09074	0.09132	0.91407	0.00672	8379.7
2.0	1000000	0.09080	0.02015	0.00028	0.09052	0.09107	0.95066	0.00581	12129.0
2.1	1000000	0.09083	0.02019	0.00028	0.09055	0.09111	0.95582	0.00663	13957.8
2.2	1000000	0.09095	0.02043	0.00028	0.09067	0.09123	1.01423	0.00427	55498.8
2.3	1000000	0.09087	0.02043	0.00028	0.09059	0.09115	0.97199	0.00588	21097.8
2.4	1000000	0.09088	0.02043	0.00028	0.09060	0.09116	0.93718	0.00885	3945.5
2.5	1000000	0.09085	0.02102	0.00028	0.09056	0.09113	0.97624	0.00485	14879.9
2.6	1000000	0.09097	0.02120	0.00029	0.09069	0.09126	0.98671	0.00958	24441.2
2.7	1000000	0.09117	0.02145	0.00029	0.09089	0.09146	0.97487	0.00784	9687.8
2.8	1000000	0.09098	0.02163	0.00029	0.09069	0.09127	1.07270	0.00543	64024.3
2.9	1000000	0.09047	0.02179	0.00029	0.09018	0.09076	1.02162	0.01259	56419.2
3.0	1000000	0.09107	0.02225	0.00029	0.09078	0.09136	0.98583	0.01627	7191.0
3.1	1000000	0.09101	0.02265	0.00029	0.09071	0.09130	0.97596	0.00792	3892.9
3.2	1000000	0.09075	0.02301	0.00030	0.09045	0.09105	0.98830	0.00929	3538.5
3.3	1000000	0.09092	0.02343	0.00030	0.09062	0.09122	1.03139	0.01042	33644.4
3.4	1000000	0.09132	0.02386	0.00030	0.09102	0.09162	0.99005	0.01413	2336.7
3.5	1000000	0.09084	0.02409	0.00030	0.09054	0.09114	0.97963	0.01016	1289.2
3.6	1000000	0.09083	0.02455	0.00031	0.09052	0.09114	0.99721	0.01045	5380.4
3.7	1000000	0.09081	0.02492	0.00031	0.09050	0.09112	1.00188	0.01244	4226.1
3.8	1000000	0.09104	0.02540	0.00031	0.09072	0.09135	0.99656	0.01401	5038.3
3.9	1000000	0.09076	0.02570	0.00031	0.09044	0.09107	0.99159	0.01814	1209.0
4.0	1000000	0.09101	0.02620	0.00032	0.09069	0.09133	1.00322	0.01474	3420.7
4.1	1000000	0.09110	0.02670	0.00032	0.09078	0.09142	0.99692	0.01190	1826.7
α=0.9									
0.8	1000000	0.09094	0.06773	0.00051	0.09043	0.09145	0.67921	0.01390	17694.5
0.9	1000000	0.09125	0.04539	0.00042	0.09083	0.09167	0.73929	0.02216	22479.5
1.0	1000000	0.09108	0.03757	0.00038	0.09070	0.09146	0.95801	0.02690	73408.0
1.1	1000000	0.09095	0.03289	0.00036	0.09059	0.09130	0.72162	0.02542	9338.5
1.2	1000000	0.09098	0.02558	0.00031	0.09067	0.09130	0.82437	0.02722	32664.2
1.3	1000000	0.09106	0.02526	0.00031	0.09075	0.09137	0.88158	0.02553	23470.9
1.4	1000000	0.09093	0.02221	0.00029	0.09064	0.09122	0.91383	0.02433	34381.6
1.5	1000000	0.09081	0.02130	0.00029	0.09053	0.09110	0.92312	0.02946	52971.9

Continued on Next Page...

β	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
1.6	1000000	0.09089	0.02060	0.00028	0.09061	0.09117	0.91551	0.03452	25030.9
1.7	1000000	0.09078	**0.01930**	0.00027	0.09051	0.09106	0.96194	0.04261	17860.3
1.8	1000000	0.09089	0.01986	0.00028	0.09061	0.09117	0.99307	0.03741	40978.2
1.9	1000000	0.09104	0.01973	0.00028	0.09077	0.09132	0.91283	0.04099	17116.5
2.0	1000000	0.09094	0.01977	0.00028	0.09066	0.09121	0.97815	0.03927	21144.5
2.1	1000000	0.09099	0.02049	0.00028	0.09071	0.09127	0.92283	0.04365	3799.6
2.2	1000000	0.09084	0.02022	0.00028	0.09056	0.09112	0.94618	0.03824	6989.4
2.3	1000000	0.09081	0.02064	0.00028	0.09053	0.09109	0.98647	0.04780	25189.5
2.4	1000000	0.09105	0.02105	0.00028	0.09076	0.09133	0.98698	0.04713	14466.3
2.5	1000000	0.09095	0.02143	0.00029	0.09066	0.09123	1.00089	0.05547	30074.1
2.6	1000000	0.09070	0.02180	0.00029	0.09041	0.09099	1.01168	0.05898	26881.3
2.7	1000000	0.09090	0.02229	0.00029	0.09061	0.09120	0.99067	0.06649	5027.1
2.8	1000000	0.09101	0.02273	0.00030	0.09071	0.09131	0.98011	0.04867	8315.9
2.9	1000000	0.09053	0.02313	0.00030	0.09024	0.09083	1.00714	0.05040	13199.1
3.0	1000000	0.09119	0.02383	0.00030	0.09088	0.09149	0.98422	0.07520	2057.6
3.1	1000000	0.09094	0.02428	0.00031	0.09064	0.09125	0.98091	0.06688	2347.9
3.2	1000000	0.09097	0.02477	0.00031	0.09066	0.09127	0.99202	0.06904	2946.3
3.3	1000000	0.09099	0.02531	0.00031	0.09068	0.09130	0.99368	0.07321	3165.0
3.4	1000000	0.09085	0.02589	0.00032	0.09054	0.09117	0.99852	0.08372	3935.7
3.5	1000000	0.09079	0.02636	0.00032	0.09047	0.09111	0.99328	0.08219	2753.3
3.6	1000000	0.09069	0.02685	0.00032	0.09037	0.09101	1.00077	0.07482	2889.3
3.7	1000000	0.09092	0.02755	0.00033	0.09060	0.09125	0.99508	0.07266	1703.6
3.8	1000000	0.09094	0.02809	0.00033	0.09061	0.09126	0.99829	0.07898	3752.3
3.9	1000000	0.09122	0.02871	0.00033	0.09089	0.09155	1.00101	0.09402	3338.0
4.0	1000000	0.09074	0.02912	0.00033	0.09040	0.09107	0.99602	0.06952	2547.6
4.1	1000000	0.09088	0.02980	0.00034	0.09054	0.09122	1.00973	0.10096	3222.8

APPENDIX C: RESULTS FOR CHAPTER 4

Table C.1. Results for dependent Exponential IS density with $c = 2$, 2A0B $\rho = 0.1$

λ_{IS}	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
0.1	1000000	0.09114	0.07258	0.00053	0.09061	0.09167	1.00031	0.0	165.5
0.2	1000000	0.09113	0.03514	0.00037	0.09076	0.09149	0.98927	0.0	4338.2
0.3	1000000	0.09100	0.02292	0.00030	0.09070	0.09130	1.35121	0.0	373509.2
0.4	1000000	0.09088	0.01706	0.00026	0.09063	0.09114	1.69979	0.0	751885.8
0.5	1000000	0.09083	0.01380	0.00023	0.09060	0.09106	1.96227	0.0	1035519.1
0.6	1000000	0.09076	0.01186	0.00021	0.09054	0.09097	0.88538	0.0	14773.7
0.7	1000000	0.09082	0.01078	0.00020	0.09062	0.09102	1.65091	0.0	831372.5
0.8	1000000	0.09082	0.01026	0.00020	0.09062	0.09101	1.70035	0.0	914619.6
0.9	1000000	0.09088	**0.01022**	0.00020	0.09068	0.09108	1.72103	0.0	958727.8
1.0	1000000	0.09086	0.01060	0.00020	0.09066	0.09106	1.70576	0.0	974251.0
1.1	1000000	0.09087	0.01146	0.00021	0.09066	0.09108	1.65736	0.0	969894.7
1.2	1000000	0.09092	0.01288	0.00022	0.09070	0.09115	1.58597	0.0	952239.8
1.3	1000000	0.09094	0.01507	0.00024	0.09070	0.09118	0.66610	0.0	35693.1
1.4	1000000	0.09100	0.01871	0.00027	0.09073	0.09127	2.47073	0.0	968228.9

Table C.2. Results for dependent Exponential IS density for 3A1B $\rho = 0.1$

c	λ_{IS}	n	$\hat{\theta}$	MSE	HL	LB	UB	AvrgW	MinW	MaxW
0.3	0.1	1000000	0.00824	0.00637	0.00016	0.00808	0.00839	1.00006	0.00000	225.64
	0.2	1000000	0.00828	0.00182	0.00008	0.00820	0.00836	0.99570	0.00000	5251.76
	0.3	1000000	0.00828	0.00093	0.00006	0.00822	0.00834	0.94345	0.00000	10563.98
	0.4	1000000	0.00830	0.00061	0.00005	0.00826	0.00835	0.93795	0.00000	45527.86
	0.5	1000000	0.00827	0.00046	0.00004	0.00823	0.00831	0.82225	0.00000	34431.24
	0.6	1000000	0.00824	0.00039	0.00004	0.00821	0.00828	0.69793	0.00000	13034.77
	0.7	1000000	0.00828	0.00036	0.00004	0.00824	0.00831	0.78086	0.00000	152483.38
	0.8	1000000	0.00823	0.00036	0.00004	0.00819	0.00827	0.78741	0.00000	142749.67
	0.9	1000000	0.00828	0.00040	0.00004	0.00824	0.00832	0.58641	0.00000	43676.59
	1	1000000	0.00826	0.00043	0.00004	0.00822	0.00830	0.53494	0.00000	51598.2
	1.1	1000000	0.00831	0.00053	0.00005	0.00826	0.00835	0.51026	0.00000	85408.9
0.5	0.1	1000000	0.00829	0.00625	0.00015	0.00814	0.00845	0.99555	0.00000	229.3
	0.2	1000000	0.00826	0.00177	0.00008	0.00818	0.00834	0.97738	0.00000	5447.4
	0.3	1000000	0.00824	0.00089	0.00006	0.00818	0.00830	0.94697	0.00000	15387.9
	0.4	1000000	0.00824	0.00058	0.00005	0.00820	0.00829	1.41506	0.00000	474904.2
	0.5	1000000	0.00827	0.00043	0.00004	0.00823	0.00831	0.84875	0.00000	44811.0
	0.6	1000000	0.00827	0.00037	0.00004	0.00823	0.00830	0.71900	0.00000	27854.5
	0.7	1000000	0.00829	0.00036	0.00004	0.00826	0.00833	0.68894	0.00000	72117.8
	0.8	1000000	0.00824	**0.00032**	0.00003	0.00820	0.00827	0.63295	0.00000	46192.4
	0.9	1000000	0.00829	0.00035	0.00004	0.00825	0.00833	0.47774	0.00000	11974.8
	1	1000000	0.00828	0.00046	0.00004	0.00824	0.00832	0.57938	0.00000	157079.2
	1.1	1000000	0.00826	0.00073	0.00005	0.00820	0.00831	0.62064	0.00000	166485.0
0.7	0.1	1000000	0.00818	0.00616	0.00015	0.00803	0.00834	1.00174	0.00000	458.4
	0.2	1000000	0.00823	0.00178	0.00008	0.00815	0.00831	0.98472	0.00000	4725.3
	0.3	1000000	0.00826	0.00090	0.00006	0.00820	0.00832	0.95901	0.00000	31207.5
	0.4	1000000	0.00826	0.00058	0.00005	0.00821	0.00831	0.87351	0.00000	43238.2
	0.5	1000000	0.00825	0.00044	0.00004	0.00821	0.00829	0.72066	0.00000	13595.8
	0.6	1000000	0.00829	0.00038	0.00004	0.00825	0.00833	0.70408	0.00000	95064.4
	0.7	1000000	0.00823	0.00033	0.00004	0.00820	0.00827	1.23412	0.00000	582731.9
	0.8	1000000	0.00828	0.00033	0.00004	0.00824	0.00831	0.60158	0.00000	56774.8
	0.9	1000000	0.00827	0.00035	0.00004	0.00823	0.00830	0.50680	0.00000	26846.3
	1	1000000	0.00826	0.00038	0.00004	0.00822	0.00830	0.48852	0.00000	66420.3
	1.1	1000000	0.00823	0.00041	0.00004	0.00819	0.00827	0.46737	0.00000	41023.0

APPENDIX D: C CODES

Here we give one of the C codes we had written. We had three files in our .NET 2003 project namely mm1.c, lcgrand.c, and the header file lcgrand.h. The whole simulation code is written in mm1.c and the other files are just used for safely Uniform(0,1) random number generating. The file mm1.c for independent Exponential IS density is given in section D.1 and the other files are given in section D.2.

D.1. C Code for Independent Exponential IS Density with Changing the Rate

```
/*** Single Server Queueing System ***/

#include <stdio.h>
#include <math.h>
#include "lcgrand.h" /* Header file for random-number generator. */

#define REP  1000000 /* Replication # for simulation  */
#define Q_LIMIT 10000 /* Limit on customer space  */
#define BUSY      1 /* Mnemonics for server's being busy */
#define IDLE      0 /* and idle. */
#define BUFFERSIZE 0 /* queue length  0,1,5,20,22 */

int   next_event_type, num_custs_delayed, num_delays_required, num_events,
      num_in_q, server_status,strm,sample,i,c1;
float area_num_in_q, area_server_status, rate_original,rate_proposal_max,mean_service,
      sim_time, time_arrival[Q_LIMIT + 1], time_last_event, time_next_event[3],
      total_of_delays,w,result,rate,term1,term2,variance,variance2,result2,min_weight,max_weight,
  indicator_qx,tot_time_for_arrv;

float * count;  /*   pointer for the result of simulation, buffer exceedance  */

FILE  *infile, *outfile,*outfile2;

void  replication(void);
void  initialize(void);
void  timing(void);
void  arrive(void);
void  depart(void);
void  report(void);
void  update_time_avg_stats(void);
float arrival_dist(float lamda);
float service_dist(float nu);
float weight(float rv);

void main()
{
    infile  = fopen("mm1_infile.txt",  "r"); /* Open input and output files. */
    outfile = fopen("mm1_outfile.txt", "w");
outfile2 = fopen("mm1_outfile2.txt", "w");

    /* Read input parameters. */
    fscanf(infile, "%f %f %d %f %f", &rate_original, &mean_service,&num_delays_required,
&rate_proposal_max);

count=& indicator_qx;         //  result of q(x) function

for(rate=rate_original;rate<=rate_proposal_max;)
{
strm=1;
result=0.0;
result2=0.0;
term1=0.0;
min_weight=1.0;
max_weight=1.0;

for(sample=1;sample<=REP;sample++)
{
```

```
replication();
result2 +=w;
result +=(*count) * w;
term1 += pow(((*count)*w),2);
if(w < min_weight)
min_weight = w;
if(w > max_weight)
max_weight = w;
}
term2 = result / REP;
variance= ((term1/REP)-(term2 * term2)) *REP /(REP -1); //var of result

fprintf(outfile2, "%.10f", (result/REP));
fprintf(outfile2, "\t%.10f", variance);
fprintf(outfile2, "\t\t\t\t%.10f", result2/REP);
fprintf(outfile2, "\t%f",min_weight);
fprintf(outfile2, "\t%f\n",max_weight);
//fprintf(outfile2, " \n variance naive = %f\n ", variance2);
++strm;
rate=rate+0.1;
}

fclose(infile);
  fclose(outfile);
fclose(outfile2);
}

/****************** replication function*****************************/

void replication(void)
{
/* Specify the number of events for the timing function. */
    num_events = 2;

/* Initialize the simulation. */
initialize();

/* Run the simulation while more delays are still needed. */

while ( c1 < num_delays_required) {

        /* Determine the next event. */

        timing();

        /* Update time-average statistical accumulators. */

        update_time_avg_stats();

        /* Invoke the appropriate event function. */

        switch (next_event_type) {
            case 1:
                arrive();
                break;
            case 2:
                depart();
                break;
        }
    }
    /* Invoke the report generator and end the simulation. */
    //report();

}

/******************** Initialization function. *******************************/

void initialize(void)
{
float y;  //variable for first arrival time

/* Initialize the simulation clock. */

    sim_time = 0.0;

/* initialize stats for exp*/
w=1.0;
indicator_qx=0.0;
c1=0;
/* Initialize the state variables. */

    server_status   = IDLE;
    num_in_q        = 0;
    time_last_event = 0.0;

    /* Initialize the statistical counters. */

    num_custs_delayed  = 0;
    total_of_delays    = 0.0;
    area_num_in_q      = 0.0;
```

```
    area_server_status = 0.0;

    /* Initialize event list.  Since no customers are present, the departure
       (service completion) event is eliminated from consideration. */
y=arrival_dist(rate);
//tot_time_for_arrv = y;  //total sim time till desired arrival number
    time_next_event[1] = sim_time + y;
    time_next_event[2] = 1.0e+30;

}

/************************************ Timing function. ***********************************/

void timing(void)
{
    int   i;
    float min_time_next_event = 1.0e+29;

    next_event_type = 0;

    /* Determine the event type of the next event to occur. */

    for (i = 1; i <= num_events; ++i)
        if (time_next_event[i] < min_time_next_event) {
            min_time_next_event = time_next_event[i];
            next_event_type     = i;
        }

    /* Check to see whether the event list is empty. */

    if (next_event_type == 0) {

        /* The event list is empty, so stop the simulation. */

        fprintf(outfile, "\nEvent list empty at time %f", sim_time);
        exit(1);
    }

    /* The event list is not empty, so advance the simulation clock. */

    sim_time = min_time_next_event;
}

/************************************* Arrival event function. *********************************/

void arrive(void)
{
    float delay;
float x;

    /* Schedule next arrival. */

x = arrival_dist(rate);         //random number for interarrival
time_next_event[1] = sim_time + x;
++c1;
if(c1<num_delays_required)
w *= weight(x); //call weight function and calculate the weight

if(c1>=num_delays_required)
time_next_event[1] = 1.0e+30;

    /* Check to see whether server is busy. */

    if (server_status == BUSY) {

        /* Server is busy, so increment number of customers in queue. */

        ++num_in_q;

if(num_in_q>BUFFERSIZE)
{
indicator_qx=1;
}

 /* Check to see whether an overflow condition exists.
if (num_in_q > Q_LIMIT) {    // The queue has overflowed, so stop the simulation.
fprintf(outfile, "\nOverflow of the array time_arrival at");
fprintf(outfile, " time %f", sim_time);
exit(2);   } // There is still room in the queue, so store the time of arrival of the
           arriving customer at the (new) end of time_arrival. */

        if(num_in_q<=BUFFERSIZE)
time_arrival[num_in_q] = sim_time;

if(num_in_q>BUFFERSIZE)
--num_in_q;

    }

    else {
```

```
        /* Server is idle, so arriving customer has a delay of zero.  (The
           following two statements are for program clarity and do not affect
           the results of the simulation.)

        delay           = 0.0;
        total_of_delays += delay;
*/

        /* Increment the number of customers delayed, and make server busy. */

        ++num_custs_delayed;
        server_status = BUSY;

        /* Schedule a departure (service completion). */

        time_next_event[2] = sim_time + service_dist(mean_service);
    }
}

/********************************** Departure event function. ********************************/

void depart(void)
{
    int   i;
    float delay;

    /* Check to see whether the queue is empty. */

    if (num_in_q == 0) {

        /* The queue is empty so make the server idle and eliminate the
           departure (service completion) event from consideration. */

        server_status      = IDLE;
        time_next_event[2] = 1.0e+30;
    }

    else {

        /* The queue is nonempty, so decrement the number of customers in
           queue. */

        --num_in_q;

        /* Compute the delay of the customer who is beginning service and update
           the total delay accumulator. */

        delay           = sim_time - time_arrival[1];
        total_of_delays += delay;

        /* Increment the number of customers delayed, and schedule departure. */

        ++num_custs_delayed;
        time_next_event[2] = sim_time + service_dist(mean_service);

        /* Move each customer in queue (if any) up one place. */

        for (i = 1; i <= num_in_q; ++i)
            time_arrival[i] = time_arrival[i + 1];
    }
}

/****************************** Report generator function. ********************************/

void report(void)
{
    /* Compute and write estimates of desired measures of performance. */
/*
   fprintf(outfile, "\n\nAverage delay in queue%11.3f minutes\n",total_of_delays / num_custs_delayed);
    fprintf(outfile, "Average number in queue%10.3f\n",(area_num_in_q / sim_time));
    fprintf(outfile, "Server utilization%15.3f\n",(area_server_status / sim_time));
    fprintf(outfile, "Time simulation ended%12.3f minutes\n", sim_time);
fprintf(outfile, " naive sim result is %d\n", naive);
fprintf(outfile, "weight of %d. sim is %.15f\n",sample, w);
*/
}

/********************* Update area accumulators for time-average  statistics. ***************************/

void update_time_avg_stats(void)
{
    float time_since_last_event;

    /* Compute time since last event, and update last-event-time marker. */

    time_since_last_event = sim_time - time_last_event;
    time_last_event       = sim_time;

    /* Update area under number-in-queue function. */
```

```
    area_num_in_q    += num_in_q * time_since_last_event;

    /* Update area under server-busy indicator function. */

    area_server_status += server_status * time_since_last_event;
}

/****************** Service dist. random variate generation function*******************/

float service_dist(float nu)
{
    /* Return an arrival_distential random variate with mean "mean". */

    return (-nu *log(lcgrand(strm)));
}

/******************** Arrival dist RV generate ******************************/

float arrival_dist(float lamda)
{
    /* Return an arrival_distential random variate with lamda "lamda". */

    return (-1/lamda) * log(lcgrand(strm));
}

/****************************** calculate weight***********************************/

float weight(float rv)
{
float func_f;
float func_g;

func_f= exp(- (rv)*rate_original)*rate_original;
func_g= exp(- (rv)*rate)*rate;

return  (func_f / func_g);
}
```

D.2. C File and Header File for Uniform Random Number Generator

We used Law and Kelton's [16] efficient Uniform(0,1) random number generator that we use those random numbers in generating interarrival times and service times besides defensive mixture distributions. We give the code below along the header file that we used in our all experiments.

```
/* Prime modulus multiplicative linear congruential generator
   Z[i] = (630360016 * Z[i-1]) (mod(pow(2,31) - 1)), based on Marse and Roberts'
   portable FORTRAN random-number generator UNIRAN.  Multiple (100) streams are
   supported, with seeds spaced 100,000 apart.  Throughout, input argument
   "stream" must be an int giving the desired stream number.  The header file
   lcgrand.h must be included in the calling program (#include "lcgrand.h")
   before using these functions.

   Usage: (Three functions)

   1. To obtain the next U(0,1) random number from stream "stream," execute
          u = lcgrand(stream);
      where lcgrand is a float function.  The float variable u will contain the
      next random number.

   2. To set the seed for stream "stream" to a desired value zset, execute
          lcgrandst(zset, stream);
      where lcgrandst is a void function and zset must be a long set to the
      desired seed, a number between 1 and 2147483646 (inclusive).  Default
      seeds for all 100 streams are given in the code.

   3. To get the current (most recently used) integer in the sequence being
      generated for stream "stream" into the long variable zget, execute
          zget = lcgrandgt(stream);
      where lcgrandgt is a long function. */

/* Define the constants. */

#define MODLUS 2147483647
#define MULT1       24112
#define MULT2       26143
```

```
/* Set the default seeds for all 100 streams. */

static long zrng[] =
{         1,
 1973272912, 281629770,  20006270,1280689831,2096730329,1933576050,
  913566091, 246780520,1363774876, 604901985,1511192140,1259851944,
  824064364, 150493284, 242708531,  75253171,1964472944,1202299975,
  233217322,1911216000, 726370533, 403498145, 993232223,1103205531,
  762430696,1922803170,1385516923,  76271663, 413682397, 726466604,
  336157058,1432650381,1120463904, 595778810, 877722890,1046574445,
   68911991,2088367019, 748545416, 622401386,2122378830, 640690903,
 1774806513,2132545692,2079249579,  78130110, 852776735,1187867272,
 1351423507,1645973084,1997049139, 922510944,2045512870, 898585771,
  243649545,1004818771, 773686062, 403188473, 372279877,1901633463,
  498067494,2087759558, 493157915, 597104727,1530940798,1814496276,
  536444882,1663153658, 855503735,  67784357,1432404475, 619691088,
  119025595, 880802310, 176192644,1116780070, 277854671,1366580350,
 1142483975,2026948561,1053920743, 786262391,1792203830,1494667770,
 1923011392,1433700034,1244184613,1147297105, 539712780,1545929719,
  190641742,1645390429, 264907697, 620389253,1502074852, 927711160,
  364849192,2049576050, 638580085, 547070247 };

/* Generate the next random number. */

float lcgrand(int stream)
{
    long zi, lowprd, hi31;

    zi     = zrng[stream];
    lowprd = (zi & 65535) * MULT1;
    hi31   = (zi >> 16) * MULT1 + (lowprd >> 16);
    zi     = ((lowprd & 65535) - MODLUS) +
             ((hi31 & 32767) << 16) + (hi31 >> 15);
    if (zi < 0) zi += MODLUS;
    lowprd = (zi & 65535) * MULT2;
    hi31   = (zi >> 16) * MULT2 + (lowprd >> 16);
    zi     = ((lowprd & 65535) - MODLUS) +
             ((hi31 & 32767) << 16) + (hi31 >> 15);
    if (zi < 0) zi += MODLUS;
    zrng[stream] = zi;
    return (zi >> 7 | 1) / 16777216.0;
}


void lcgrandst (long zset, int stream) /* Set the current zrng for stream
                                          "stream" to zset. */
{
    zrng[stream] = zset;
}


long lcgrandgt (int stream) /* Return the current zrng for stream "stream". */
{
    return zrng[stream];
}
```

And here we give the very little header file called lcgrand.h used for just definitons:

```
/* The following 3 declarations are for use of the random-number generator
   lcgrand and the associated functions lcgrandst and lcgrandgt for seed
   management.  This file (named lcgrand.h) should be included in any program
   using these functions by executing
       #include "lcgrand.h"
   before referencing the functions. */

float lcgrand(int stream);
void  lcgrandst(long zset, int stream);
long  lcgrandgt(int stream);
```

REFERENCES

1. Hesterberg, T., 1995, “Weighted Average Importance Sampling and Defensive Mixture Distributions”, *Technometrics*, Vol. 37, No.2,May.

2. Bassamboo, A., S. Juneja and A. Zeevi, 2006, “On the inefficiency of state-independent importance sampling in the presence of heavy tails”, *Elsevier*, Operations Research Letters 0167 6377, February 10.

3. Yuan, C. and M. J. Druzdzel, 2005, “Importance sampling algorithms for Bayesian networks: Principles and performance”, *Elsevier*, Mathematical and Computer Modelling 43 (2006) 1189 1207, May 4.

4. Campioni, L., R. Scardovelli and P. Vestrucci, 2004, “Biased Monte Carlo optimization: the basic approach”, *Elsevier*, Reliability Engineering and System Safety 87 (2005) 387 394, September 13.

5. Ross, K. W. and J. Wang, 1996, “Implemantion of Monte Carlo integration for the analysis of product-form queueing networks”, *Elsevier*, Performance Evaluation 29 (1997) 273 292, May 30.

6. Kuruganti, I. and S. Strickland, 1997, “Optimal importance sampling for Markovian systems with applications to tandem queues”, *Elsevier*, Mathematics and Computers in Simulation 44 (1997) 61 79.

7. Suziki, S. and K. Nakagawa, 2000, “Performance Evaluation of an ATM Switch with Back Pressure by “IS” Simulation”, *Scripta Technica*, Electronics and Communications in Japan, Part 1, Vol.83, No.9, 2000.

8. Xiao, G., Z. Li and T. Li, 2006, “Dependability estimation for non-Markov consecutive-k-out-of-n: F repairable systems by fast simulation”, *Elsevier*, Reliability Engineering and System Safety 92 (2007) 293-299, May 30.

9. Hurtado, J. E., 2005, “Filtered importance sampling with support vector margin: A powerful method for structural reliability analysis”, *Elsevier*, Structural Safety (2006), December 8.

10. Chen, Y., 2005, "Another look at rejection sampling through importance sampling", *Elsevier*, Statistics and Probability Letters (2005), February 19.

11. Bekaert, P., M. Sbert and Y. D. Willems, 2000 "Weighted Importance Sampling Technique for Monte Carlo Radiosity", *11th Eurographics Workshop on Rendering*, Brno, Czech Republic.

12. Evans, M. and T. Swartz, 2000, *Approximating integrals via Monte Carlo and Deterministic Methods*, Oxford Statistical Science Series, New York.

13. Bratley, P., B. L. Fox and L. E. Schrage, 1987, *A Guide to Simulation*, Second Edition, Springer Verlag, New York.

14. Fishman, G. S., 2006, *A First Course in Monte Carlo*, Duxbury Advanced Series, USA.

15. Kalos, M. H. and P. A. Whitlock, 1986, *Monte Carlo Methods*, Volume I:Basics, John Wiley & Sons, Canada.

16. Law, A. M. and W. D. Kelton, 2000, *Simulation Modeling and Analysis*, Mc Graw Hill, Singapore.

17. Ross, S. M., 1997, *Simulation*, Academic Press, California.

18. Sminchisescu, C. and B. Triggs, 2002 "Hyperdynamics Importance Sampling", *ECCV*.

REFERENCES NOT CITED

Agarwal, S., R. Ramamoorthi, S. Belongie and H. W. Jensen, 2004, "Structured Importance Sampling of Environment Maps", *ACM SIGGRAPH Conference Proceedings*, Computer Graphics.

Dupuis, P. and H. Wang, 2005, "Dynamic Importance Sampling for Uniformly Recurrent Markov Chains", *The Annals of Applied Probability*, Institute of Mathematical Statistics.

http://www.tcm.phy.cam.ac.uk/ ajw29/thesis/node17.html

http://www.maths.tcd.ie/ chas/node51.html

http://rkb.home.cern.ch/rkb/AN16pp/node132.html