Efficient Sampling Algorithm for Optimization under Uncertainty

BY

NISHANT SATEJ DIGE
B.E. Production Engineering
University of Pune, 2012

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of Science in Industrial Engineering
in the Graduate College of the
University of Illinois at Chicago, 2016

Chicago, Illinois

Defense Committee:

Dr. Urmila Diwekar, Chair and Advisor
Dr. Houshang Darabi
Dr. Michael Scott

ABSTRACT

Uncertainty is a part of a real world optimization problem. Computational speed is critical in optimizing large scale stochastic problems. The major bottleneck in solving large scale stochastic optimization problems is the computational intensity of scenarios or samples. This research proposes a novel sampling technique which takes above mentioned problem. This thesis analyzes existing and novel sampling techniques by conducting large scale experiments with different functions. The sampling techniques which were analyzed are Monte Carlo Sampling (MCS), Latin Hypercube Sampling (LHS), Hammersley Sequence Sampling (HSS), Latin Hypercube-Hammersley Sequence Sampling (LHS-HSS), Sobol Sampling, and the proposed novel technique which is Latin Hypercube-Sobol Sampling (LHS-SOBOL). It was found that HSS performs better up to 40 uncertain variables, Sobol up to 100 variables, LHS-HSS up to 250 variables, and LHS-SOBOL for large scale uncertainties which was tested for 800 variables. Thus, by analyzing the results of this work we can conclude that LHS-HSS can be used for uncertainties from 2 to 100 variables, and LHS-SOBOL for larger than 100 variables.

Keywords: Sampling, Quasi Sequence Sampling, Sobol Sampling, Optimization under uncertainty, LHS-Sobol sampling, stochastic supply chain network problem

ACKNOWLEDGEMENT

I would like to thank Dr. Urmila Diwekar my advisor for this thesis work for her constant guidance and support. Her motivation and guidance has helped me learn new things and specially developed a desire for knowledge. Thank you Dr. Houshang Darabi and Dr. Michael Scott for being on my defense committee. Finally I would like to thank my parent for continued support since my childhood without whom I would not be here.

Table of content

LIST OF FIGURES

LIST OF TABLES

Chapter 1. INTRODUCTION

The problem of optimizing distribution through a supply chain has been long studied. In their seminal work, Geoffrion and Graves (1974) optimize the location of distribution facilities between plants and customers using Bender's decomposition. Thomas and Griffin (1996), Vidal and Goetschalckx (1997), and Klose and Drexl (2005) present reviews of other works that focus on deterministic production and distribution planning (David et al., 2015).

The uncertain nature of many elements of the supply chain, however, often means that deterministic approaches lead to suboptimal results. Stochastic programming, rather than linear programming, may better capture the realities of a business in which future conditions are uncertain (Sen and Higle 1999). A number of authors have addressed uncertainty in planning a single tier of the supply chain, such as production planning and scheduling or transportation decisions, as Sahinidis (2004) and Mula et al. (2006) have reviewed. However, most of the articles in supply chain management under uncertainty literature solve small scale problems. In real world, the problems tend to be large with many number of uncertain variables. For example, consider the USG problem given below.

1. 1. USG Supply chain Problem definition:

USG is a building materials and solutions manufacturer in North America and the problem we are talking about focuses on Durock product line. Under this product line, several items are produced at 3 manufacturing locations in United States. These products are passed to the customers throughout North America either directly or via one of the warehouses. Figure 1 shows the schematic of the network. There are 3 manufacturing plants, 54 warehouses, and 1500 customer locations.

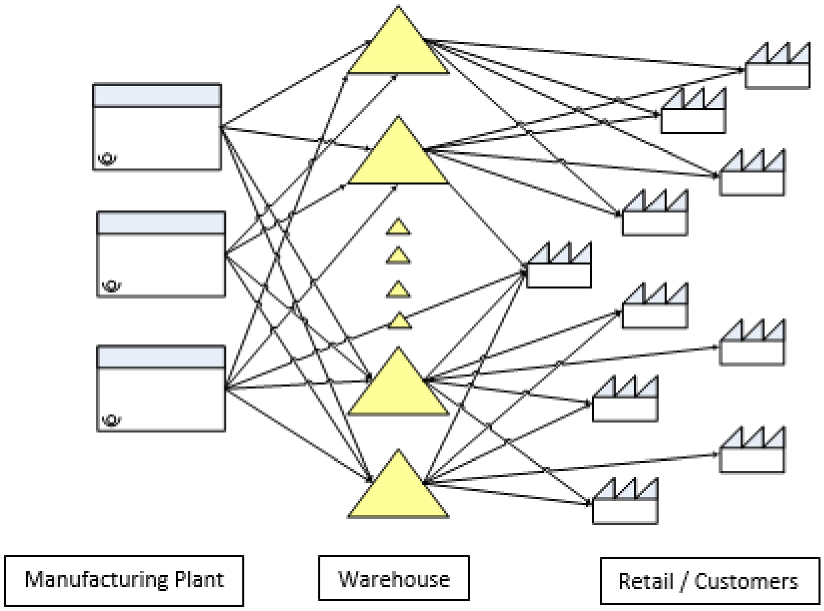

Figure 1. Supply chain network

Currently the sourcing decisions are computed by solving large scale linear programing models (LP) with an objective to minimize the total delivery cost. This delivery cost comprises of production, freight and handling cost. While achieving this objective of minimizing the cost, we consider constraints like capacity, demand and warehouse balance at respective manufacturing plants, customer locations and warehouses. The linear programming problem is very large as there are 1500 customer locations, to which products manufactured at 3 plants are shipped either directly or via one of the 54 warehouses. In addition, number of items produced at these locations are between thirty and forty, along with the choice for the inbound shipment to the warehouse which is via rail or truck. In the deterministic problem of USG, there are 90000 decision variables. However, the cost parameters and demand for each location is uncertain. From the data provided by USG, it can be seen that there are 6000 uncertain variables in this problem. Recently David (David A., 2015; David et al., 2015) used the existing data to reduce the uncertain variables to 800. In order to solve this problem, they converted it into a chance constrained formulation as shown below.

Problem formulation (David A., 2015):

$$Min\ Z = E\left[\begin{array}{c}\sum_i \sum_k \sum_l \left(P_i + f_{ki} + \frac{H_k}{x_t}\right) X_{kil} + \sum_i \sum_j \sum_l \left(P_i + f_{jir} + \frac{H_{jr}}{x_r}\right) X_{rjil} + \\ \sum_i \sum_j \sum_l \left(P_i + f_{jit} + \frac{H_{jt}}{x_t}\right) X_{tjil} + \sum_k \sum_j \sum_l \left(f_{kjt} + \frac{H_{kt}}{x_t}\right) X_{tkjl}\end{array}\right] \text{...... (1)}$$

Subjected to constraints:

$$p(\sum_i X_{kil} + \sum_k X_{kjtl} - d_j) \geq 0.95 \qquad \text{...... (2)}$$

$$p = Probability\ that\ each\ demand\ comstraint\ is\ met$$

$$\sum_k X_{kil} + \sum_j (X_{jrl} + X_{jtl}) \leq c_i\ \forall\ i, l \qquad \text{...... (3)}$$

$$\sum_i (X_{jirl} + X_{jitl}) \leq \sum_k X_{kjtl} \qquad \text{...... (4)}$$

$$i = 1, \dots 3; j = 1, \dots 1500; k = 1, \dots 54; l = 1, \dots 37;\ r = rail, t = truck$$

David et al. (2015) solved this problem by converting the chance constraint into equivalent deterministic constraint. However, chance constraint programming and this transformation is applicable to stable distributions only. The data shows that some of the distributions in the problem are not stable and hence need better handling. A generalized approach for solving stochastic optimization problems is shown in Figure 2. In deterministic optimization, only the optimization loop is required where decisions are changed and the effects of these changed decisions on the objective function and constraints can be noted by running the model. The optimizer then checks the optimality criterion and the loop continues, if the criterion is not satisfied till an optimal solution is found. For stochastic programming or stochastic optimization problem, it involves an additional internal loop where uncertainties are handled using sampling methods. The major bottleneck in such problems is the stochastic sampling loop, which calculates the estimates of probabilistic objective function and constraints. For the USG problem, we need to efficiently handle 800 dimensional uncertainties with this sampling loop. This is the focus of our current endeavors.

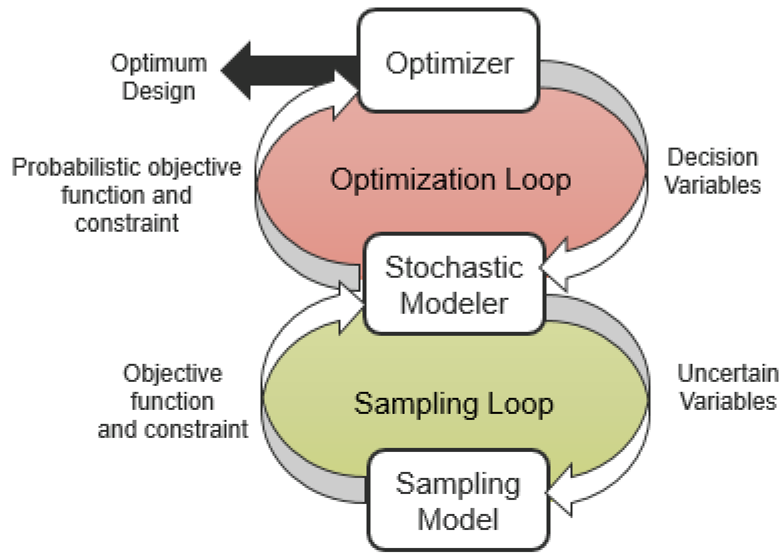

Figure 2. Stochastic optimization framework

1. 2. <u>Literature review of sampling techniques:</u>

Sampling is a statistical procedure that involves the selection of a finite number of individuals to represent and infer knowledge about a population of concern. Sampling techniques are used in a wide range of science and engineering applications; they are of basic importance in computational statistics, in implementation of probabilistic algorithms, and in related problems of statistical computing that have a stochastic ingredient (Ulas and Diwekar, 2007).

Monte Carlo sampling (MCS) is the most simple and widely used sampling technique. It is a numerical method based on a pseudorandom random generator to approximate solution for a variety of mathematical problem (Metropolis and Ulam, 1949). The name Monte Carlo originates from the city in Morocco which is famous for casinos and randomness surrounded with it. Example of the pseudorandom number generator is the linear congruential generator developed by Lehmer (Lehmer, 1949). In general, Monte Carlo needs large number of samples for obtaining probabilistic information about the output. In most applications, the actual relationship between successive points in a sample has no physical significance, hence, randomness of the sample for approximating a uniform distribution is not critical

(Knuth, 1973). Once it was apparent that the uniformity properties are critical to the design of sampling techniques, constrained or stratified sampling technique became appealing (Morgan and Henrion, 1990). Latin Hypercube sampling (LHS) is one of the well-known stratified sampling techniques (Iman and Conover, 1982). In this sampling technique, each variable distribution is divided into number of equi-probable zones and sample is taken randomly from each zone (Iman and Shortencarier, 1984). These single dimensional samples for each variables are then paired randomly. It has been found that LHS performs much better than MCS. However, LHS is not multi-dimensionally uniform, a desirable property for sampling. Quasi Monte Carlo methods based on low discrepancy sequences like Hammersley sequence, show better uniformity property. Hammersley sequence sampling (HSS) technique proposed by Kalagnanam and Diwekar (1997) is found to be an efficient sampling technique, for solving problems of small dimensional uncertainties. It has been found that HSS shows spurious correlations when applied to high dimensional problems. In order to circumvent this problem, Kocis and Whiten (1997) proposed Leaped HSS and HSS-RR2. However, both these techniques distort the k-dimensional uniformity of the HSS resulting in lower efficiency. In 2002, Wang et al. (2002) coupled LHS with HSS to derive a better sampling technique, which tries to achieve uniformity in one dimension as well as K-dimensions. Financial literature requires high dimensional random numbers for uncertainty evaluations. Sobol is very popular in financial literature. However Sobol faces the same problem of spurious correlations in higher dimension (Chi et al., 2005). To circumvent this problem scrambled Sobol sampling was introduced by Chi et al.(2005). However scrambled Sobol disturbs the k-dimensional uniformity.

In this work, we propose a new sampling technique based on Latin Hypercube and Sobol sampling techniques. We compare all six different sampling techniques with 216000 experiments to study the behavior of the proposed and current sampling techniques.

Chapter 2. CURRENT SAMPLING TECHNIQUES

2. 1. Introduction

In chapter 1 we briefly described various sampling techniques which are mentioned below.

- Monte Carlo Sampling technique
- Stratified Sampling (Latin Hypercube sampling technique)
- Hammersley Sampling technique
- Leaped Hammersley technique
- LHS-Hammersley Sampling technique
- Sobol Sampling technique
- Scrambled Sobol Sampling technique.

This chapter concentrates on providing the details about how samples are generated from each sampling technique, what are the advantages and disadvantages of these sampling techniques starting with the Monte Carlo Sampling.

2. 2. Monte Carlo Sampling (MCS):

Monte Carlo is the most widely used sampling technique since 19th century. Monte Carlo sampling generates random samples using pseudorandom numbers (Metropolis and Ulam, 1949). One of the popular pseudorandom number generator is the Linear Congruential Generator algorithm by Lehmer (1949). Once pseudorandom number generator generates random numbers between 0 and 1 for each uncertain variable, specific values of the sample are generated by inverting these random numbers on cumulative distribution function (CDF) for each variable.

One of the major advantage of Monte Carlo technique is that the error bounds are not dependent on the dimension of the problem. Error ‘ε’ is inversely proportional to the square root of number of samples ‘n’. Another major advantage is that the bounds are probabilistic, which is not achieved by any other method.

$$\varepsilon = O/\sqrt{n}$$

However there are two important properties for any sampling technique; randomness and uniformity. In most applications, the actual relationship between successive points in a sample has no physical significance, hence, randomness of the sample for approximating a uniform distribution is not critical (Knuth, 1973). Figure 3 shows Monte Carlo samples for 2 uncertain variables with uniform distributions between 0 and 1 for each. It can be seen that the samples are not covering the square uniformly and there are clusters observed. This decreases the efficiency of MCS and thus requires large samples.

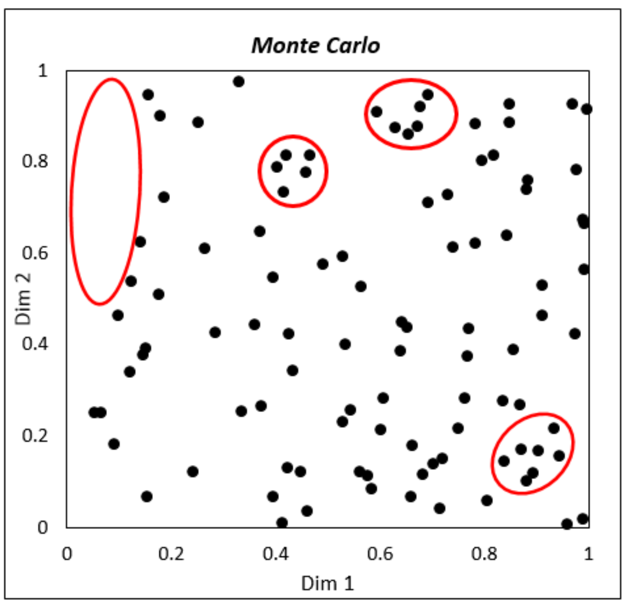

Figure 3. Clustering effect in Monte Carlo

Once it is apparent that the uniformity properties are critical to the design of sampling techniques, constrained or stratified sampling becomes appealing (Morgan and Henrion, 1990). Latin hypercube is one such sampling and is described below.

2. 3. Latin Hypercube Sampling (LHS):

LHS aims at evenly spreading the samples in the distribution by dividing the distribution into equi-probable zones (Iman and Shortencarier, 1984) as shown in Figure 4 and samples are drawn randomly from each zone. Number of zones equals number of samples. LHS shows good one dimensional uniformity as shown in Figure 5. For good one dimensional uniformity, the samples have to be closer to the 45 degree line.

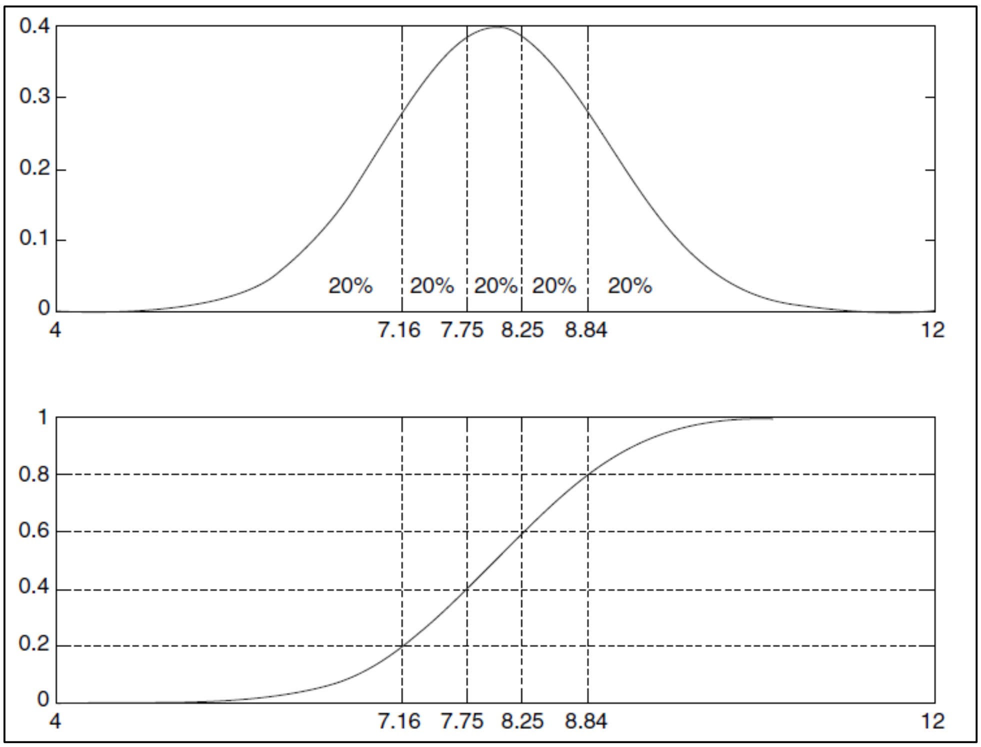

Figure 4. Normal distribution divided into equiprobable zone (Diwekar, 2008)

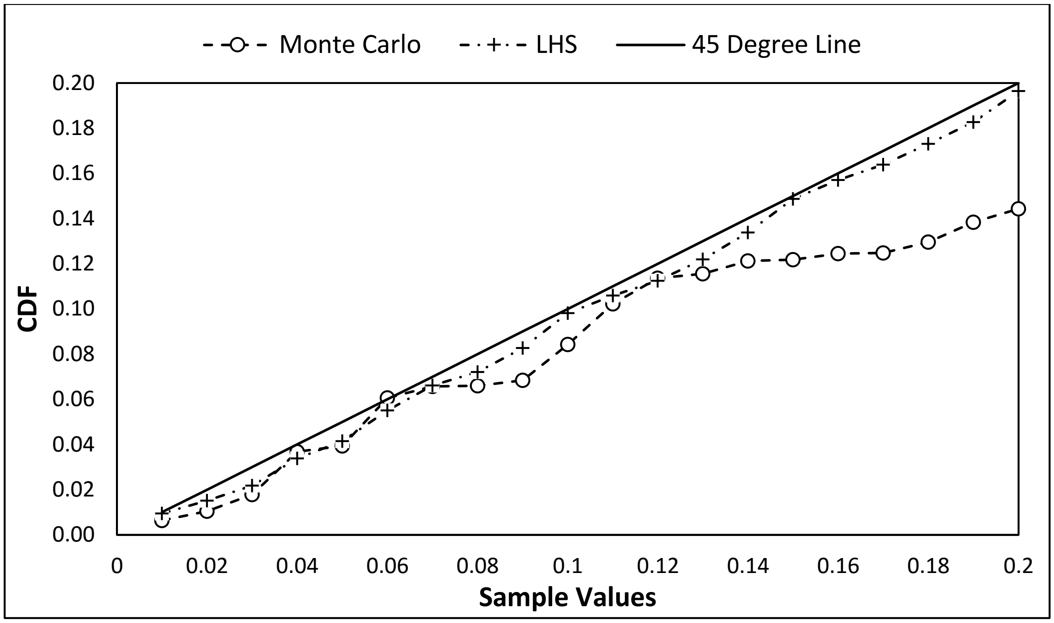

Figure 5. Comparing one dimensional uniformity of LHS and MCS

For multi-dimensional problems, samples are extracted individually and randomly paired to form the sample set. Due to this random pairing, uniformity in one dimension is lost for multi-dimension sample set (refer Figure 6).

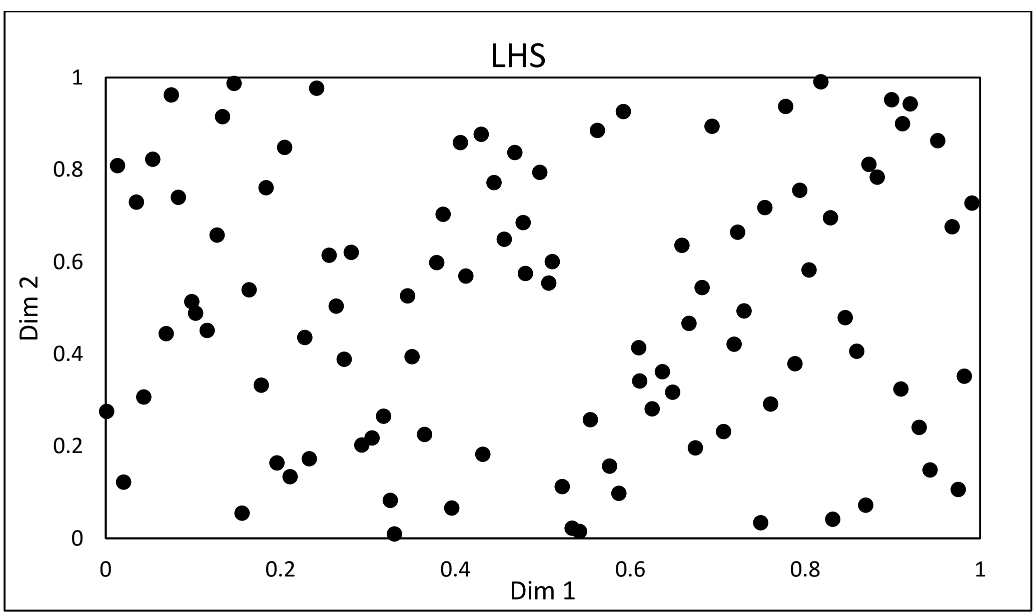

Figure 6. 100 two dimensional samples generated from LHS

Example:

Let U_1 has uniform distribution from 0-5 and U_2 has uniform distribution from 0-20. Using LHS we first generate samples by dividing U_1 and U_2 into equi-probable zone and take random points from these zones. These samples and ranks are shown in Table 1.

Rank	**U_1**
1	0.8
2	1.4
3	2.6
4	3.2
5	4.9

Rank	**U_2**
1	3.2
2	6.1
3	11.3
4	12.8
5	18.4

Table 1. Equi-probable samples and their ranks for LHS

In the second step these ranks are randomly paired and the samples are arranged accordingly as shown in Table 2 below.

Ranks Paired (U_1, U_2)	
	2
2	5
3	1
4	4
5	3

Data Set (U_1, U_2)	
0.8	6.1
2.6	18.4
3.2	3.2
1.4	12.8
4.9	11.3

Table 2. Random paring the samples

Thus multi-dimensional uniformity is lost due to random pairing. In order to overcome this drawback Quasi-Monte Carlo sampling techniques are proposed. Quasi Monte Carlo methods use different low discrepancy sequence that perform significantly better than crude Monte Carlo sampling. Some well-known low discrepancy sequences are Hammersley, Halton, Sobol, Niederreiter and Faure. These Quasi Monte Carlo techniques are used for its multi-dimensional uniformity properties.

2.4. Hammersley Sequence Sampling:

This low discrepancy sequence algorithm is based on inverse prime number radix function for generating so called Hammersley points. These Hammersley points are then inverted on CDF to generate the actual sample set. Each prime number radix corresponds to each uncertain parameter. Following is the algorithm generating Hammersley points:

Any integer n can be written in radix R notation as:

$$n = n_m n_{m-1} \ldots . n_2 n_1 n_0 \qquad \ldots\ldots (5)$$

$$n = n_0 + n_1 R + n_2 R^2 + \ldots n_m R^m \qquad \ldots\ldots (6)$$

A unique number between 0 & 1 called the inverse radix number is constructed by reversing the order of digits of n, around the decimal point as follows:

$$\Phi_r(n) = n_o R^{-1} + n_1 R^{-2} + \ldots . + n_m R^{m+1} \qquad \ldots\ldots (7)$$

Hammersley points on k-dimensional hypercube are given by the following sequence:

$$X_K(n) = 1 - Z_K(n) \qquad \ldots\ldots (8)$$

Where, $$Z_K(n) = (\, n/N \,;\, \Phi_{R1}(n);\, \Phi_{R2}(n);\, \Phi_{R3}(n);\, \ldots . \,;\, \Phi_k(n)\,) \qquad \ldots\ldots (9)$$

Following is an example of generating 30 Hammersley points for 2 Dimensional data set, for this example k = 2, and first prime, hence R1 = 2 in equation 9. Number of samples = N=30.

$$Z_2(n) = (\, n/30 \,;\, \Phi_{R1}(n))$$

		2^5	2^4	2^3	2^2	2^1					
		32	16	8	4	2					
Sr. No. (n)	***Sr. No. Integer in Binary***	***Binary Representation***					***Inverse Radix {Φr(n)}***	***Inverse Radix Point ZK(n)***	***n/N***	***Hammersley Point***	
										Dim 1	***Dim 2***
1	1					1	0.5	0.5	0.033333	0.033333	0.5
2	10				1	0	0.25	0.75	0.066667	0.066667	0.75
3	11				1	1	0.75	0.25	0.1	0.1	0.25
4	100			1	0	0	0.125	0.875	0.133333	0.133333	0.875
5	101			1	0	1	0.625	0.375	0.166667	0.166667	0.375
6	110			1	1	0	0.375	0.625	0.2	0.2	0.625
7	111			1	1	1	0.875	0.125	0.233333	0.233333	0.125
8	1000		1	0	0	0	0.0625	0.9375	0.266667	0.266667	0.9375
9	1001		1	0	0	1	0.5625	0.4375	0.3	0.3	0.4375
10	1010		1	0	1	0	0.3125	0.6875	0.333333	0.333333	0.6875
11	1011		1	0	1	1	0.8125	0.1875	0.366667	0.366667	0.1875
12	1100		1	1	0	0	0.1875	0.8125	0.4	0.4	0.8125
13	1101		1	1	0	1	0.6875	0.3125	0.433333	0.433333	0.3125
14	1110		1	1	1	0	0.4375	0.5625	0.466667	0.466667	0.5625
15	1111		1	1	1	1	0.9375	0.0625	0.5	0.5	0.0625
16	10000	1	0	0	0	0	0.03125	0.96875	0.533333	0.533333	0.96875
17	10001	1	0	0	0	1	0.53125	0.46875	0.566667	0.566667	0.46875
18	10010	1	0	0	1	0	0.28125	0.71875	0.6	0.6	0.71875
19	10011	1	0	0	1	1	0.78125	0.21875	0.633333	0.633333	0.21875
20	10100	1	0	1	0	0	0.15625	0.84375	0.666667	0.666667	0.84375
21	10101	1	0	1	0	1	0.65625	0.34375	0.7	0.7	0.34375
22	10110	1	0	1	1	0	0.40625	0.59375	0.733333	0.733333	0.59375
23	10111	1	0	1	1	1	0.90625	0.09375	0.766667	0.766667	0.09375
24	11000	1	1	0	0	0	0.09375	0.90625	0.8	0.8	0.90625
25	11001	1	1	0	0	1	0.59375	0.40625	0.833333	0.833333	0.40625
26	11010	1	1	0	1	0	0.34375	0.65625	0.866667	0.866667	0.65625
27	11011	1	1	0	1	1	0.84375	0.15625	0.9	0.9	0.15625
28	11100	1	1	1	0	0	0.21875	0.78125	0.933333	0.933333	0.78125
29	11101	1	1	1	0	1	0.71875	0.28125	0.966667	0.966667	0.28125
30	11110	1	1	1	1	0	0.46875	0.53125	1	1	0.53125

Table 3. 30 two dimensional Hammersley points

The sample points are generated by inverting Hammersley points on respective CDF (Uniform 0-1) for dimension 1 and 2 are shown in Table 3. This Figure 7 shows that Hammersley sequence sampling provides better uniformity for multi-dimensional problem.

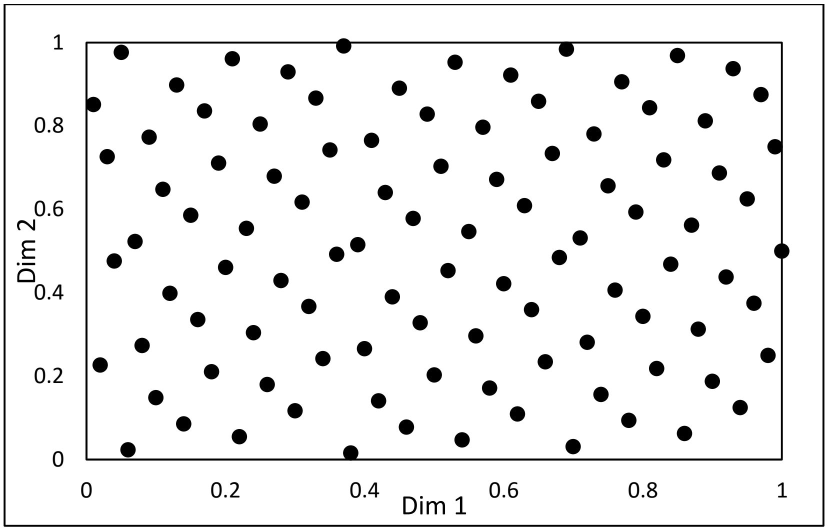

Figure 7. HSS points in 2 dimension

Major disadvantage of HSS is that as the number dimensions increase the distance between prime numbers increase, and inverse radix for these large prime numbers become smaller and smaller resulting in spurious correlations for variables more than 40 as shown in Figure 8.

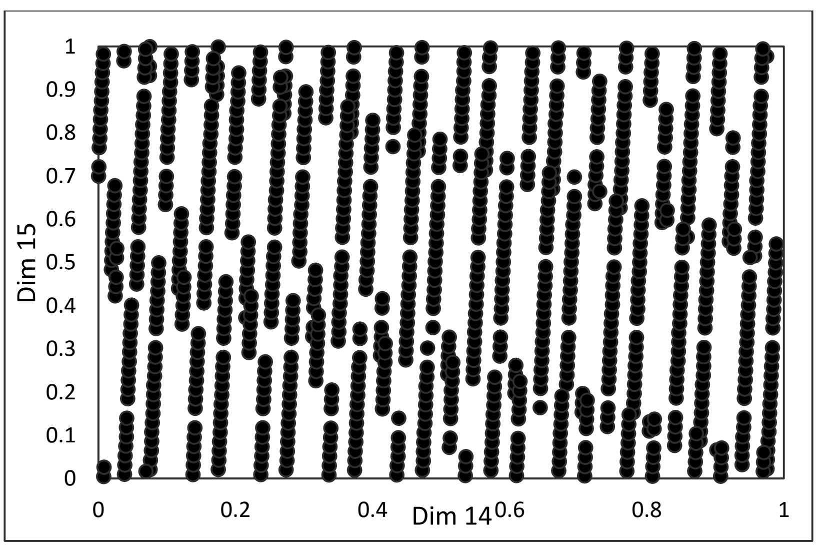

Figure 8. 1000 sample points using HSS for dimension 14 and 15

2. 5. Leaped Hammersley

Leaped Hammersley sampling is developed to overcome these spurious correlations for higher dimension (Kocis and Whiten, 1997).In this sampling technique we generate large sample points and select the samples with leap.

Generalized Hammersley inverse radix function:

$$\boldsymbol{\Phi_{r1}(n) = \sum_i a_i\,(j, n) * b_j{}^{-i-1}} \qquad \text{...... (10)}$$

As the prime number increases, denominator becomes larger and the number which represents sample integer tends to get smaller and smaller. In order to avoid this, below equation is proposed by introducing leap to the sample integer.

Leaped Hammersley inverse radix function:

$$\boldsymbol{\Phi_{r1}(nL) = \sum_i a_i\,(j, n) * b_j{}^{-i-1}} \qquad \text{...... (11)}$$

In equation 11, 'L' represents leap. Following example shows 20 points generated by HSS for dimension 3 and using leap we only select few of these as actual samples. (Please see table 3)

Sr. No.	Hammersley Points Dimension 3
1	0.666666667
2	0.333333333
3	0.888888889
4	0.555555556
5	0.222222222
6	0.777777778
7	0.444444444
8	0.111111111
9	0.962962963
10	0.62962963
11	0.296296296

12	0.851851852
13	0.518518519
14	0.185185185
15	0.740740741
16	0.407407407
17	0.074074074
18	0.925925926
19	0.62962963
20	0.259259259

Table 4. 20 points generated by HSS for dimension 3

Now from these HSS points we select 4 samples with a leap of 5 as shown in table 5.

Leap of 5	Leaped Hammersley Sample points
5	0.222222222
10	0.62962963
15	0.740740741
20	0.259259259

Table 5. Sample points with leap of 5 for dimension 3

However, these leaping distorts the K-dimensional uniformity property.

2. 6. Sobol Sequence Sampling:

Sobol sequence sampling is developed by Ian Sobol (Sobol et al., 1967). In this sequence we generate direction vectors which are dependent on primitive polynomial and the degree of polynomial depends on dimension of the problem. Primitive polynomial is the polynomial which cannot be factorized further. This sampling is explained below.

Primitive polynomial of degree d, all the coefficients A_1 to A_{d-1} are either 1 or 0 is given below.

$$\boldsymbol{P = X^d + A_1\, X^{d-1} + \cdots + A_{d-1} X + 1} \qquad \text{...... (12)}$$

Direction vector in dimension j are generated by recursive equation given below for $\mathbf{i > d}$ and the initial direction vectors i.e. for $\mathbf{i < d}$ are generated by selecting and number between 0 & 2^d.

$$\boldsymbol{V_i^j = \frac{M_i}{2^i}} \quad \text{...... (13)}$$

$$\boldsymbol{M_i = 2^1 A_1 M_{i-1} \oplus 2^2 A_2 M_{i-2} \oplus \dots\dots\dots \oplus 2^{d-1} A_{d-1} M_{i-d+1} \oplus 2^d M_{i-d} \oplus M_{i-d}} \quad \text{..... (14)}$$

$$\boldsymbol{\oplus \; is \; bitwise \; exclusive \; or \; operation}$$

Thus after generating direction vectors, Sobol number can be generated as follows.

$$\boldsymbol{x_n^j = b_1 v_i^j \oplus b_2 v_2^j \oplus \dots\dots \oplus b_w v_w^j} \quad \text{...... (15)}$$

$$\boldsymbol{n = \sum_{i=0}^{w} b_i 2^i}$$

XOR operation:

Input		***Output***
A	***B***	
0	0	0
1	0	1
0	1	1
1	1	0

Table 6. Exclusive OR operation commands

Example for generating Sobol points for dimension 3:

Identify primitive polynomial for dimension 3, i.e. it should have degree 3.

Primitive Polynomial = $\boldsymbol{X^3 + X + 1}$

M_1 = 1 (Odd integer between 0 & 2^1)

M_2 = 3 (Odd integer between 0 & 2^2)

M_3 = 7 (Odd integer between 0 & 2^2)

From Recurrent equation for $\mathbf{i > 3}$

$$M_i = 4M_{i-2} \oplus 8M_{i-3} \oplus M_{i-3} \qquad \text{...... (16)}$$

Always start with last term in forming recursive equation (Equation 16) and the equation should have same number of terms as the degree of polynomial. Thus,

$M_4 = 4M_2 \oplus 8M_1 \oplus M_1 = 12 \oplus 8 \oplus 1 = 1100 \oplus 1000 \oplus 0001 = 0101 \text{ (i.e. 5)}$

$M_5 = 4M_3 \oplus 8M_2 \oplus M_2 = 28 \oplus 24 \oplus 3 = 11100 \oplus 11000 \oplus 00011 = 00111 \text{ (i.e. 7)}$

$M_6 = 4M_4 \oplus 8M_3 \oplus M_3 = 20 \oplus 56 \oplus 7 = 010100 \oplus 111000 \oplus 000111 = 101011 \text{ (43)}$

$M_7 = 4M_5 \oplus 8M_4 \oplus M_4 = 28 \oplus 40 \oplus 5 = 011100 \oplus 101000 \oplus 000101 = 110001 \text{ (49)}$

From $\boldsymbol{M_i}$ we can get $\boldsymbol{V_i}$ for dimension 3 as;

$$\boldsymbol{V_1 = 0.1}$$

$$\boldsymbol{V_2 = 0.11}$$

$$\boldsymbol{V_3 = 0.111}$$

$$\boldsymbol{V_4 = 0.0101}$$

$$\boldsymbol{V_5 = 0.00111}$$

$$\boldsymbol{V_6 = 0.101011}$$

$$\boldsymbol{V_7 = 0.0110001}$$

Sobol points for dimension 3 can be generated as follows:

$\boldsymbol{x_1^3 = 1 = 1 * V_1 = 0.1 = 0.5}$

$\boldsymbol{x_2^3 = 10 = 0 * V_1 \oplus 1 * V_2 = 0.00 \oplus 0.11 = 0.11 = 0.75}$

$\boldsymbol{x_3^3 = 11 = 1 * V_1 \oplus 1 * V_2 = 0.10 \oplus 0.11 = 0.01 = 0.25}$

$\boldsymbol{x_4^3 = 101 = 1 * V_1 \oplus 0 * V_2 \oplus 1 * V_3 = 0.100 \oplus 0.111 = 0.011 = 0.375}$

And so on till the number of samples required.

Using Gray Code we can quickly generate Sobol Points.

Binary number to gray code:

In generating Gray Code we keep the first term same and perform XOR operator to generate remaining terms.

Thus 1st (Gray Code) = 1st (Binary) ***XOR*** 2nd (Binary)

2nd (Gray Code) = 1st (Binary) ***XOR*** 2nd (Binary)

3rd (Gray Code) = 2nd (Binary) ***XOR*** 3rd (Binary) and so on.

Example:

Binary	**1**	**0**	**0**	**1**	**1**
Gray Code	**1**	**1**	**0**	**1**	**0**

Table 7. Binary to gray code conversion

With gray codes generated we can easily write Sobol sequence algorithm as:

$$\boldsymbol{n = \sum_{i=0}^{w} b_i 2^i}$$

$$\textit{\textbf{Integer}}\ \boldsymbol{n = b_1 b_2 b_3 \ldots = g_1 g_2 g_3 \ldots}$$

$$\boldsymbol{x_1^d = g_1 V_1 \oplus g_2 V_2 \oplus g_3 V_3 \ \ldots}$$

$$\boldsymbol{x_{n+1}^d = x_n^d \oplus V_c}$$

Where $\mathrm{b_c} = \text{rightmost zero bit binary representatio}$

Generated Sobol points are inverted on to the CDF to get actual sample point. Sobol sequence sampling has better multi-dimensional uniformity (Figure 9), but correlations have been observed for higher dimensions above 100 using this sequence (Figure 10).

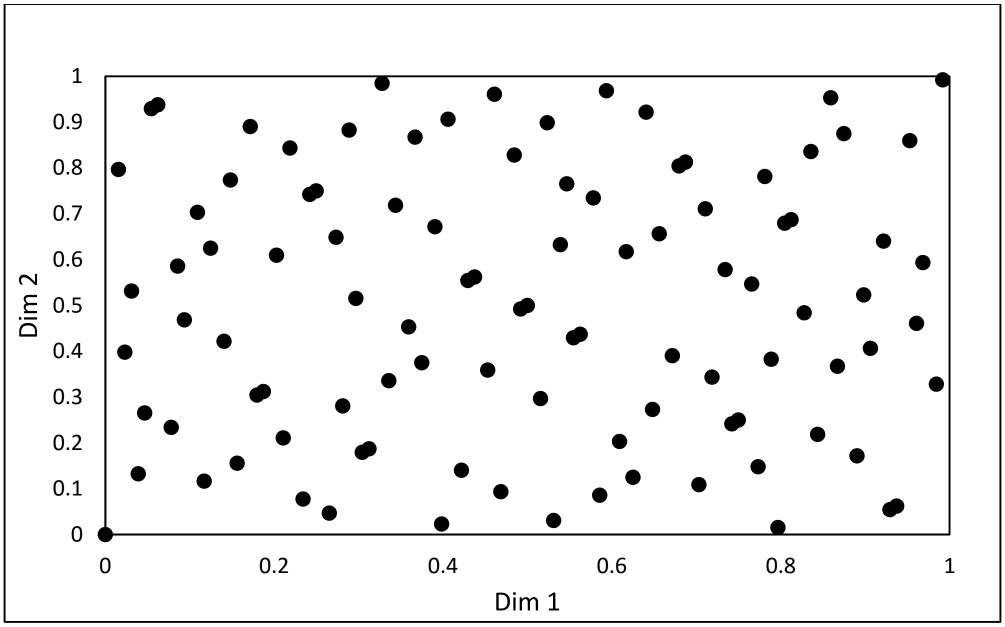

Figure 9. 100 samples using Sobol sampling for Dimension 1 and 2

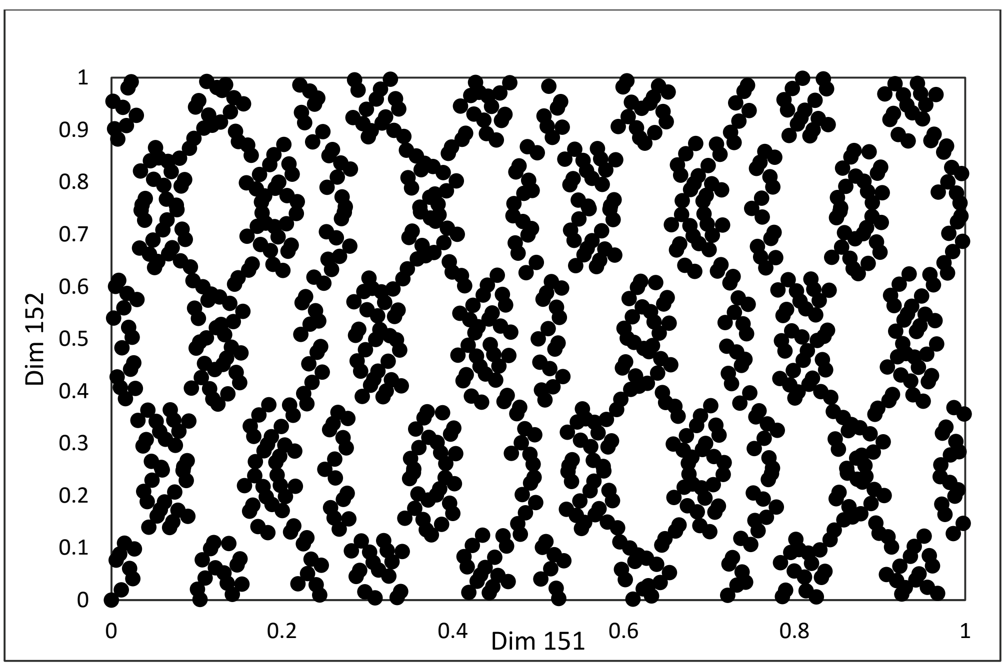

Figure 10. 1000 samples using Sobol sampling for dimension 151 and 152

2. 7. Scrambled Sobol:

In order to eliminate spurious correlations for higher dimensions Sobol sampling, scrambled Sobol sampling was introduced. In this technique binary integers are randomly scrambled to remove the correlations. The scrambling can be done in two ways, either scrambling one integer or multi-integer scrambling. A novel algorithm for multi-integer scrambling was shown by (Chi et al., 2005). They have used pure linear permutation for scrambling the binary integers prior to getting Sobol point from recursive equation. This technique is largely dependent on random permutations and if the permutations are not done properly correlations can be observed.

2.8. LHS-Hammersley Sequence Sampling (LHS-HSS):

LHS-HSS is developed by (Wang et al., 2002). In this Sampling technique we generate samples for each dimension using LHS sampling technique. Unlike in LHS where pairing is done randomly here we use Hammersley points to rank the LHS samples and pairing is done using these ranks. With this sampling technique one dimensional as well as multi-dimensional uniformity is achieved.

Example:

Here are 5 points generated by LHS for 2 dimension U_1 & U_2, uniform distribution from 0-1.

U_1	Ranks U_1	U_2	Ranks U_2
0.8	1	3.2	1
1.4	2	6.1	2
2.6	3	11.3	3
3.2	4	12.8	4
4.9	5	18.4	5

Table 8. Samples and ranks for LHS-HSS

Ranking these sample points using Hammersley Points as shown in table 9:

HSS points dimension 1	Rank 1
0.2	1
0.4	2
0.6	3
0.8	4
1	5

HSS Point dimension 2	Rank 2
0.5	3
0.75	4
0.25	1
0.875	5
0.375	2

Table 9. Corresponding HSS points for each dimension

Sample set generated using this ranking method are given in Table 10.

U_1	U_2
0.8	11.3
1.4	18.4
2.6	3.2
3.2	6.1
4.9	12.8

Table 10. Two dimensional sample set for LHS-HSS

LHS-HSS has shown good convergence properties up to 20 variables (Wang et al., 2002) and is likely to fail for large scale uncertainties like the problem presented in chapter 1. Therefore, we propose a novel sampling technique based on LHS and Sobol sampling. This is presented in next chapter.

Chapter 3. NOVEL SAMPLING TECHNIQUE LHS-SOBOL

In order to solve this supply chain problem with large number of uncertainties, we developed a new sampling technique LHS-Sobol by combining LHS sampling technique and Sobol sampling technique. This new sampling is expected to have one dimensional uniformity of LHS and k-dimensional uniformity of Sobol. In this technique we first draw samples for each uncertain parameter using LHS sampling technique thus preserving one-dimensional uniformity. This is achieved by using rank matrices of Sobol samples instead of pairing the LHS samples randomly. In order to achieve multi-dimensional uniformity we pair those using Sobol sequence points. Sobol sequence needs primitive polynomials and direction vectors. In this work, we use primitive polynomial generator proposed by Bratley and Fox (1986) and extended to higher dimension (up to 1111) by Joe and Kuo (2008). Figure 11 presents the flow chart to generate LHS-Sobol sampling. The procedure is illustrated using an example given below.

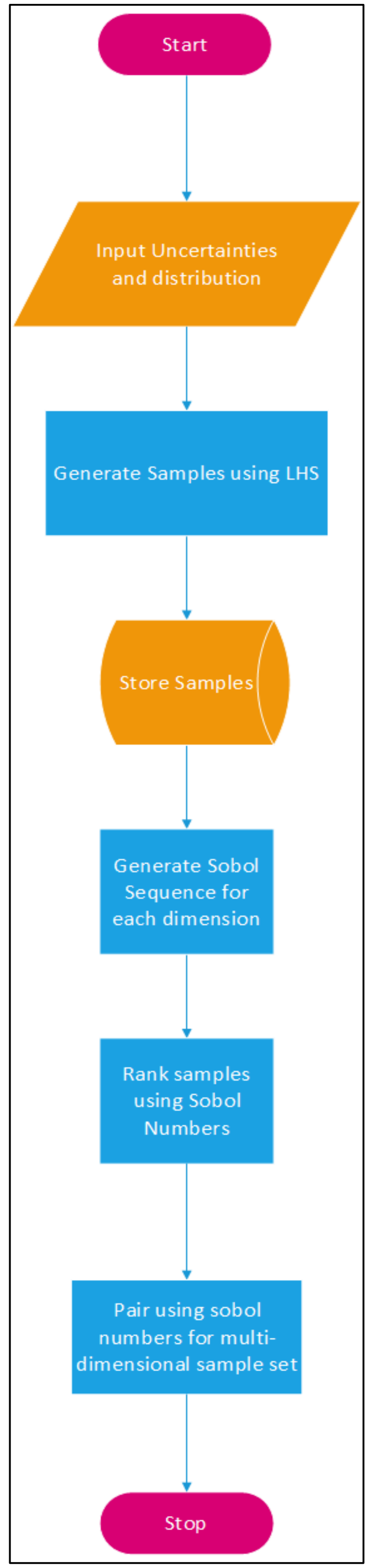

Figure 11. LHS-Sobol flow Chart

3. 1. LHS-Sobol generation steps and example:

Step 1: Let us consider 3 uncertain parameters be as follows:

U1 = Uniform Distribution (5 – 10)

U2 = Uniform Distribution (15 – 25)

U3 = Uniform Distribution (1 – 10)

Step 2: We plan to generate 10 samples. Following are samples generated using first step of Latin Hypercube where we divide the probability density function of each variables into 10 equi-probable strata and take random points from each strata.

U1	U2	U3
5.052407	15.10481	1.094332
5.6718	16.3436	2.209239
6.028591	17.05718	2.851464
6.762964	18.52593	4.173336
7.177369	19.35474	4.919264
7.691069	20.38214	5.843925
8.468701	21.9374	7.243663
8.756406	22.51281	7.761529
9.163712	23.32742	8.494681
9.921371	24.84274	9.858466

Table 11. Samples generated using LHS for each of the dimensions

Step 3: Considering ranks according to Sobol sequence.

Sobol U1	Sobol Rank U1	Sobol U2	Sobol Rank U2	Sobol U3	Sobol Rank U3
0.5	5	0.5	6	0.5	5
0.75	8	0.25	3	0.75	8
0.25	3	0.75	8	0.25	2
0.375	4	0.375	5	0.625	7
0.875	9	0.875	10	0.125	1
0.625	6	0.125	2	0.375	4
0.125	1	0.625	7	0.875	10
0.1875	2	0.3125	4	0.3125	3
0.6875	7	0.8125	9	0.8125	9
0.9375	10	6.25E-02	1	0.5625	6

Table 12. Each dimension of LHS samples ranked with Sobol points

Step 4: Pairing the samples in Table 11 using ranks of the samples in Table 12.

U1	Ranks U1	U2	Rank U2	U3	Rank U3
8.468701	5	24.84274	6	4.919264	5
8.756406	8	20.38214	3	2.851464	8
6.028591	3	16.3436	8	7.761529	2
6.762964	4	22.51281	5	5.843925	7
5.052407	9	18.52593	10	1.094332	1
7.691069	6	15.10481	2	9.858466	4
9.163712	1	21.9374	7	4.173336	10
5.6718	2	17.05718	4	2.209239	3
7.177369	7	23.32742	9	8.494681	9
9.921371	10	19.35474	1	7.243663	6

Table 13. 3 dimensional sample set using LHS-Sobol

This is the way we generate 3 dimensional sample set for using LHS-Sobol sampling technique.

Chapter 4. RESULTS AND DISCUSSION

This chapter presents the analysis of six sampling techniques including the novel sampling technique proposed in Chapter 3. In the first part of the sampling analysis, we check the one dimensional and multi-dimensional uniformity of various sampling techniques for different dimensions. In the latter part of the chapter we compare efficiencies of various sampling techniques using experiments with different functions and dimensions.

4. 1. One dimensional Uniformity:

Uniformity plays a vital role while approximating a distribution by finite samples. One dimensional and multi-dimensional uniformity properties are very important for solving large scale stochastic optimization problem. It has been shown that LHS-HSS and LHS has good one-dimensional uniformity (Wang et al., 2004). In Figure 12 we present the one-dimensional uniformity of Sobol and LHS-Sobol. It can be seen that LHS-SOBOL has good one-dimensional uniformity than Sobol or Monte Carlo.

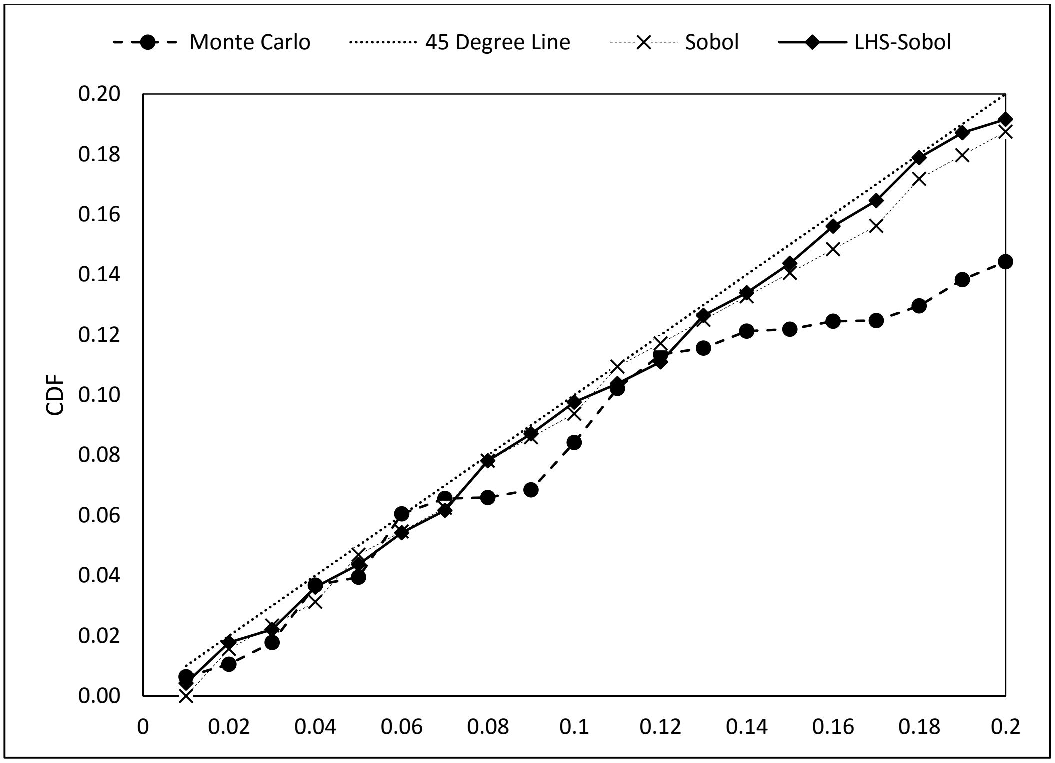

Figure 12. Comparing one dimensional uniformity for LHS-Sobol, Sobol and MCS

4. 2. Multi-Dimensional Uniformity:

Figure 13 qualitatively shows sample set generated for two dimension using different sampling techniques and are plotted in unit square. From this it is clear that Quasi Monte Carlo sampling have better uniformity compared to Monte Carlo and Latin Hypercube Sampling for multi-dimensional approximation. LHS uses random pairing once each samples are generated with one-dimensional uniformity. Due to such random paring clusters of samples are observed in certain areas preventing uniformity.

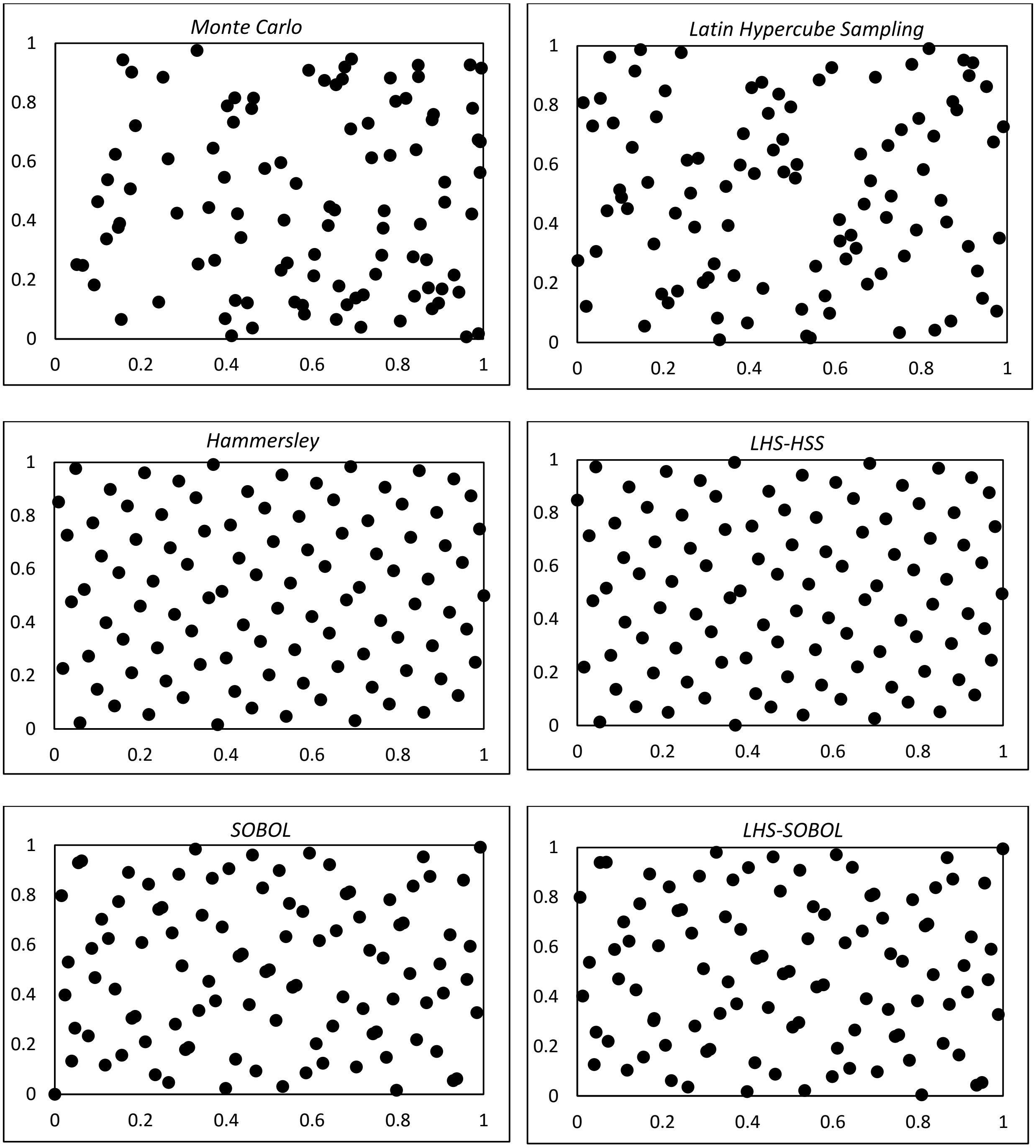

Figure 13. Two dimensional 100 sample points using different sampling techniques

4. 3. Higher Dimensional Correlations

It has been found that the quasi Monte Carlo methods like Hammersley and Sobol show spurious correlations when there no correlations exist. For HSS this occurs as early as dimension 15-16 as shown in Figure 14. However, these correlations start having effects after dimension 40 as number of correlations starts increasing as shown in Figure 14. In SOBOL the correlations start appearing as early as dimension 45 (Figure 15) and as the number of dimensions increase above 100, the effect correlations start having effect on convergence.

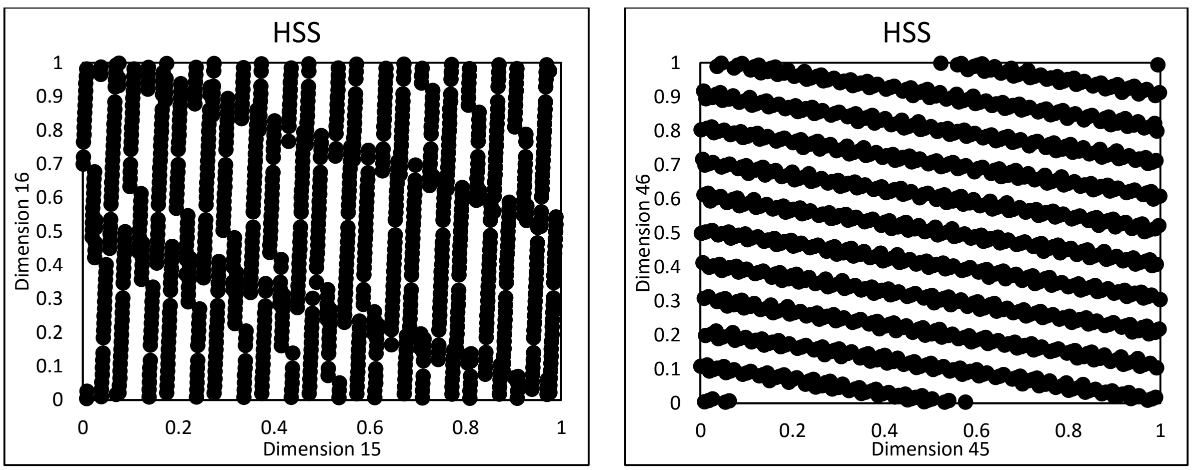

Figure 14. 1000 sample points for dimension 15, 16 and for dimension 45, 46 using HSS

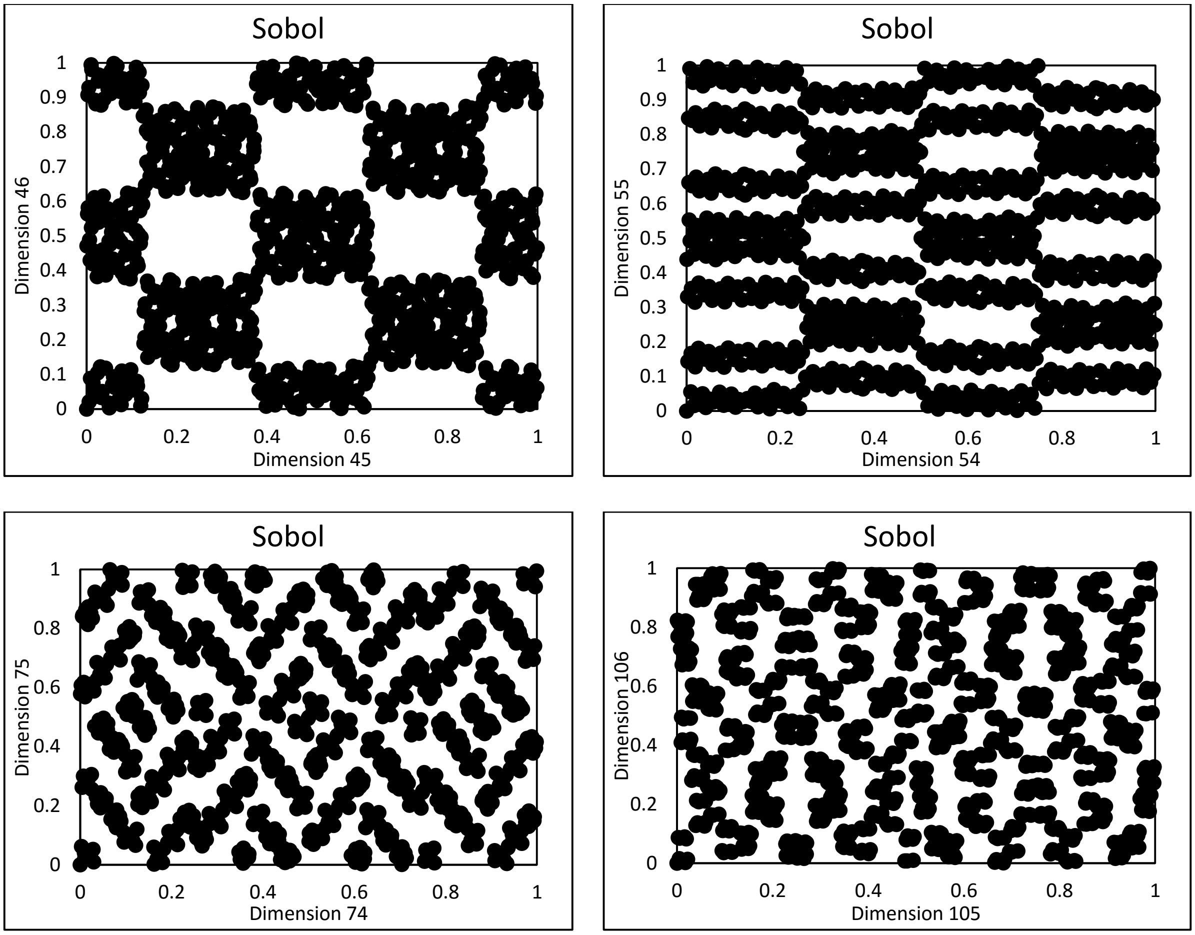

Figure 15. 1000 sample points for dimension 45-46, 54-55, 74-75 and 105-106 using Sobol sampling

LHS-Sobol breaks these correlations shown in Sobol (Figure 16) and hence likely to have better convergence above 100 uncertain variables.

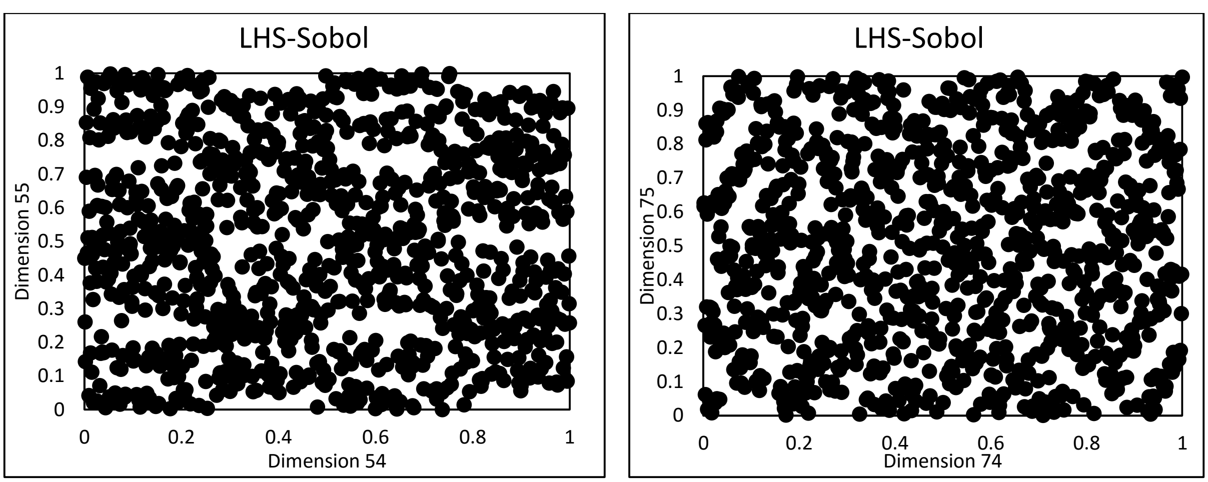

Figure 16. 1000 sample points for dimension 54-55 and 74-75 using LHS-Sobol

4.4 Efficiency and Convergence of various Sampling Techniques

In this section we have compared the performance of various sampling technique namely LHS-Sobol, LHS-HSS, Sobol, Hammersley, Latin Hypercube Sampling and Monte Carlo sampling by propagating samples generated through these sampling techniques for a set of i-input uncertain variables (Xi) through test functions (Y = f (X1, X2, X3, …., Xi). The four test functions are given below. We start with 20 uncertain variables all uniformly distributed between 0 and 1. We increase number dimensions to 40, 100, 250 and lastly 800 uncertain variables. The convergence of each sampling technique to the true values of mean, variance, and 0.95 fractile representing the tail end of output distribution is noted.

Test function:

1. *Linear additive function* $Y = \sum_i X_i$
2. *Multiplicative function* $Y = \prod_i X_i$
3. *Quadratic Function* $Y = \sum_i X_i^2$
4. *Exponential Function* $Y = \sum_i [X_i * \exp X_{i+1}]$

True values of these quantities are determined by propagating large sample set (50,000 samples) of Monte Carlo sampling. Once the true values are determined we set error limit and the performance of different sampling technique is compared by calculating the number of samples required to converge within these error limits. The error limits for mean is 0.1% up to dimension 250 and for 800 dimension it is 0.01%. We have increased the precision as for higher dimension the changes is observed beyond third decimal figure. For variance we have taken 1% error limit for 20 and 40 dimension, 2% error limit for 100 and 250 dimension and 5% error limit for 800 dimension. It is very difficult to converge variance so as the dimension increase we have increased the error bounds. For 0.95 fractile we have 1% error limit for dimensions 100, 250 and 800, for 20 and 40 dimension the values eventually converge to zero so we set error limit of 0.001. Given that there are 4 functions and number of dimensions varying from 20 to 800, 6 sampling techniques, and values of mean, variance, and fractile, this results in 216000 experiments. We considered the graphs of exponential function (Function 4) to illustrate the results first.

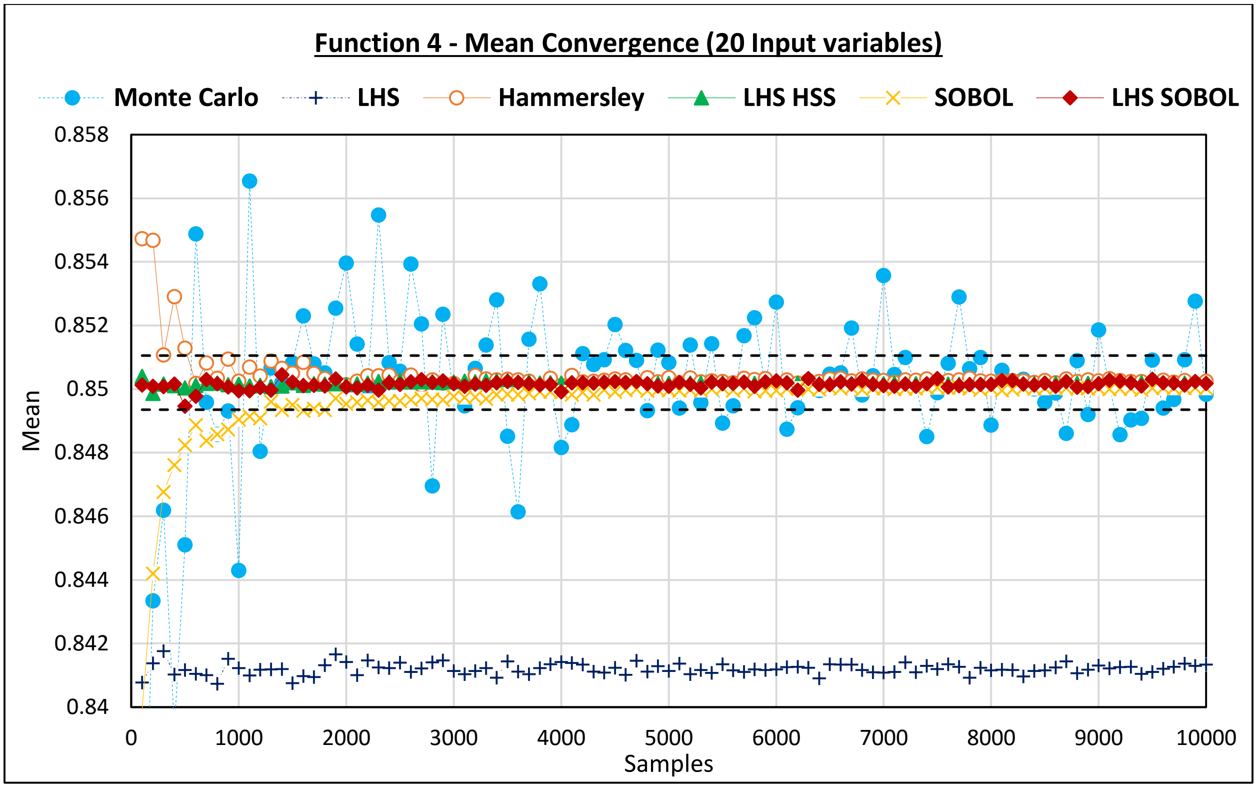

Figure 17. Mean Vs. sample size for exponential function with 20 input variables for different sampling techniques

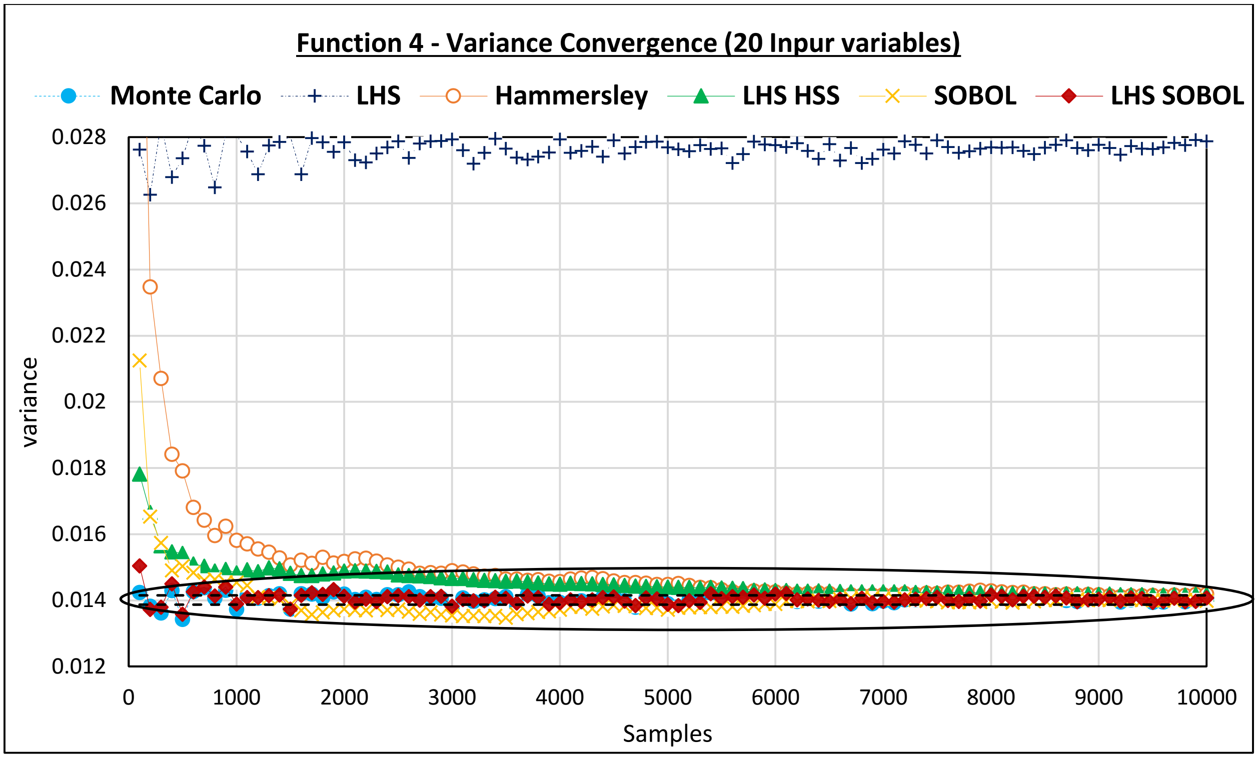

Figure 18. Variance Vs. sample size for exponential function with 20 input variables for different sampling techniques

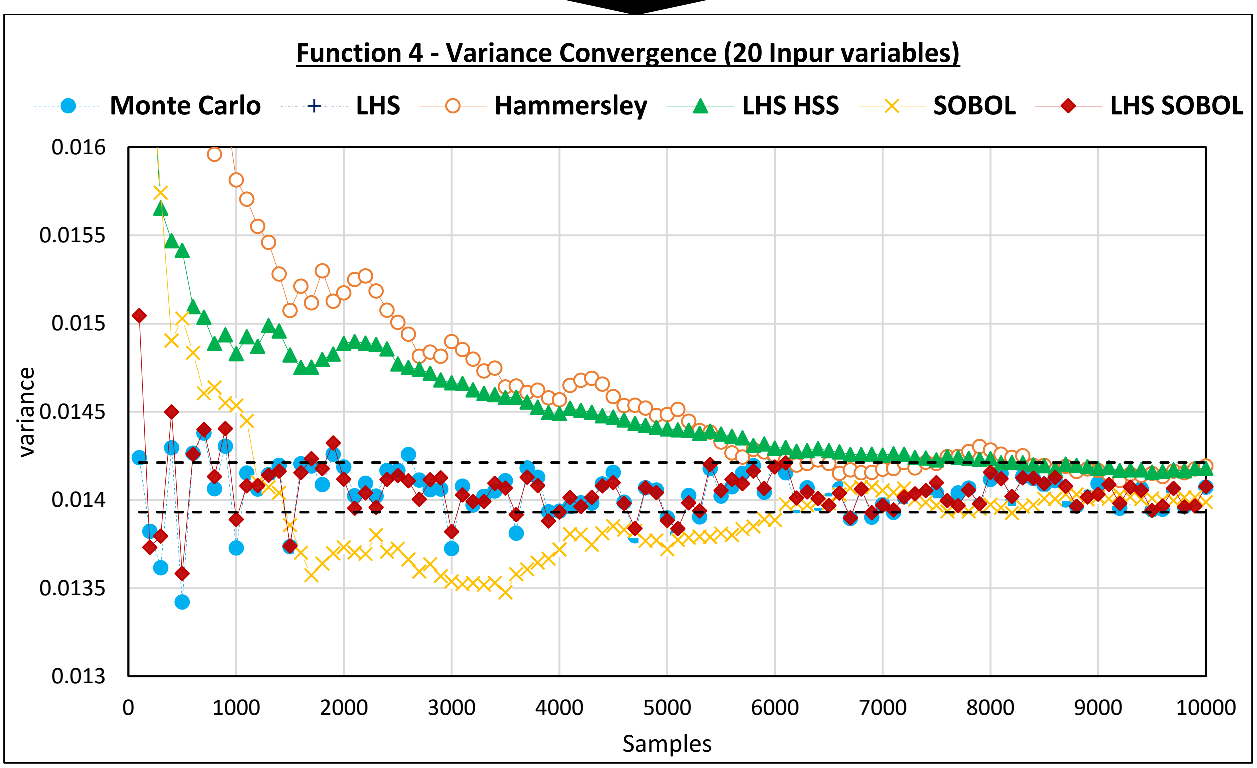

Figure 19. Zoomed in figure 18

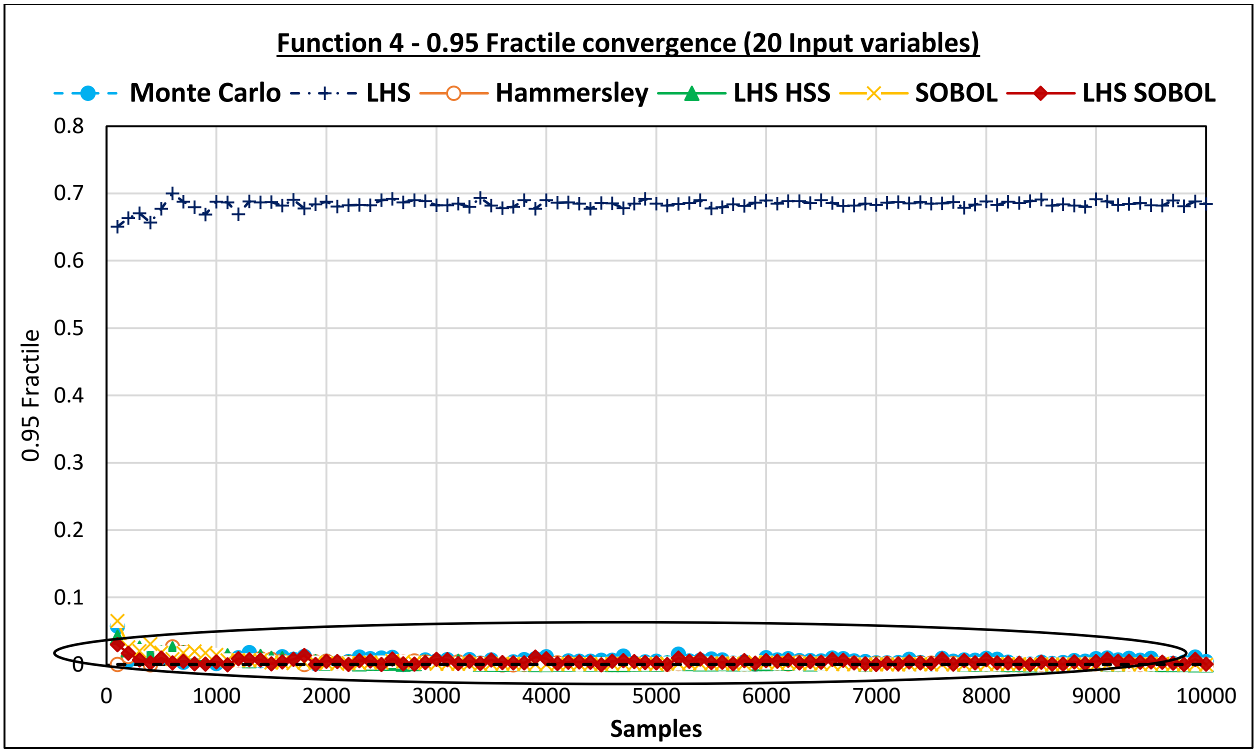

Figure 20. Fractile Vs. sample size for exponential function with 20 input variables for different sampling techniques

Zooming in

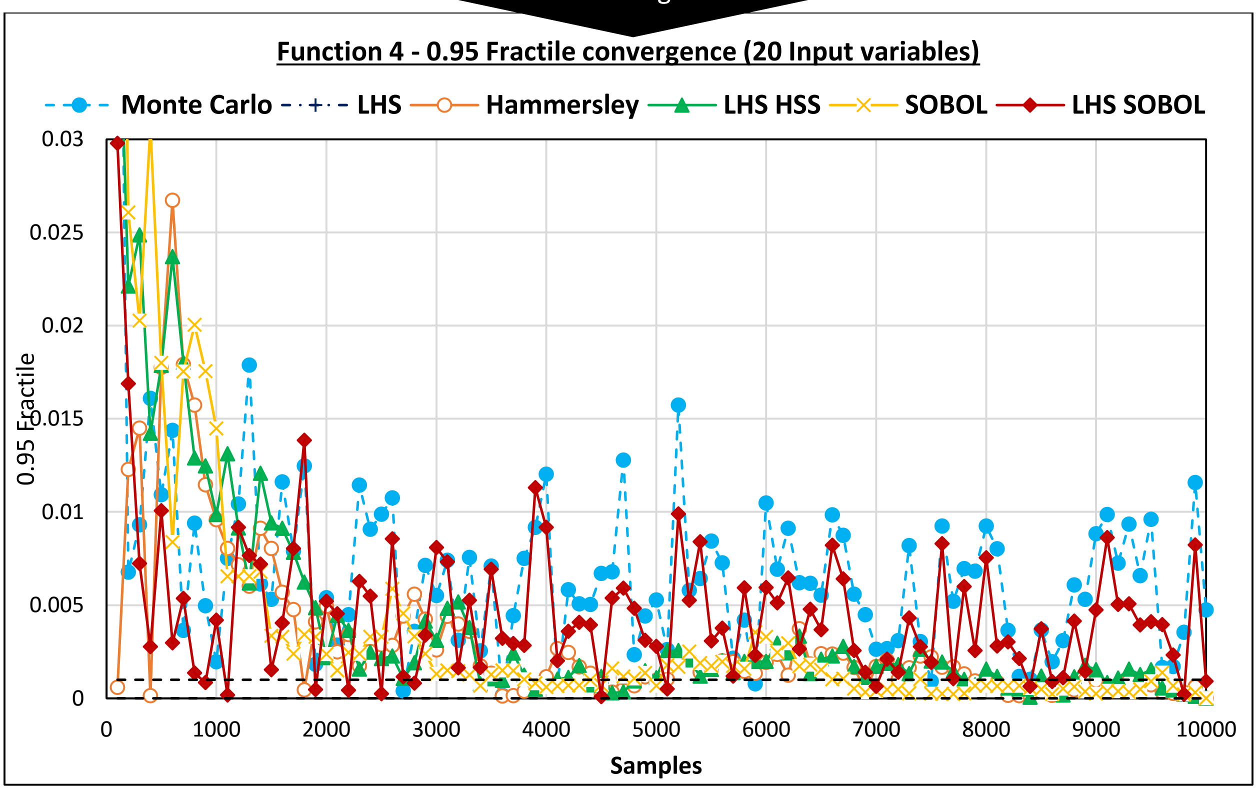

Figure 21. Zoomed in figure 20

Figure 17 to Figure 21 show mean, variance and 0.95 fractile convergence results for various sampling techniques respectively, for 20 uncertain variables. Figure 17 shows that the mean obtained by MCS shows fluctuations beyond the error band even for 10000 samples showing MCS not converged for 10000 samples. HSS and SOBOL did much better than MCS but LHS-HSS converged faster than any other sampling.

Variance convergence shown in Figure 18 reflects that, LHS techniques variance converges to completely different value which is out of the error band. HSS requires relatively large samples compared to LHS-HSS, SOBOL and LHS-SOBOL.

After looking at 0.95 fractile convergence as in Figure 20, we can see that LHS lies way outside the error band. Figure 21 shows that MCS and LHS-Sobol did not converge within the error limits for as large as 10000 samples. LHS-HSS and HSS have converged but slightly outside error limit whereas Sobol converged within the error limit for for 0.95 fractile.

Collectively looking at Mean, variance and fractile convergence it is observed that LHS-HSS performs consistently better compared to other sampling techniques for 20 uncertain variables.

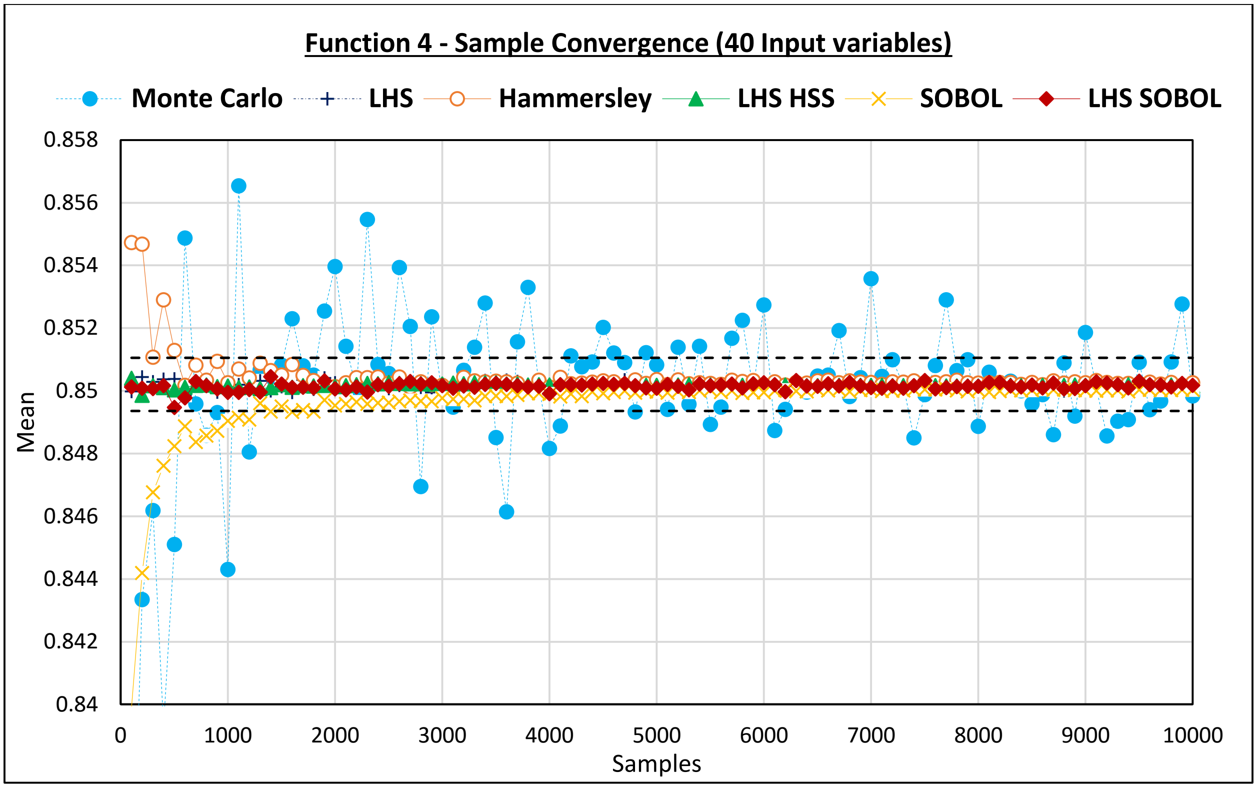

Figure 22. Mean Vs. sample size for exponential function with 40 input variables for different sampling techniques

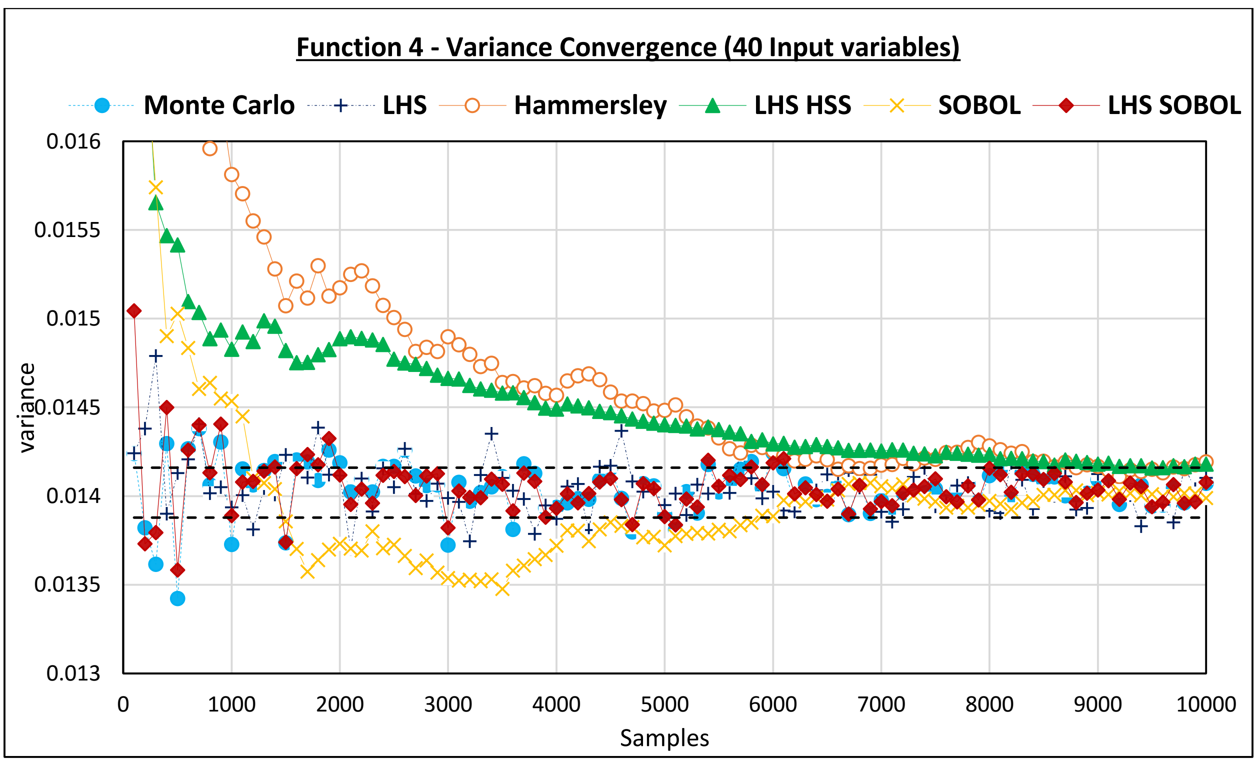

Figure 23. Variance Vs. sample size for exponential function with 40 input parameters for different sampling techniques

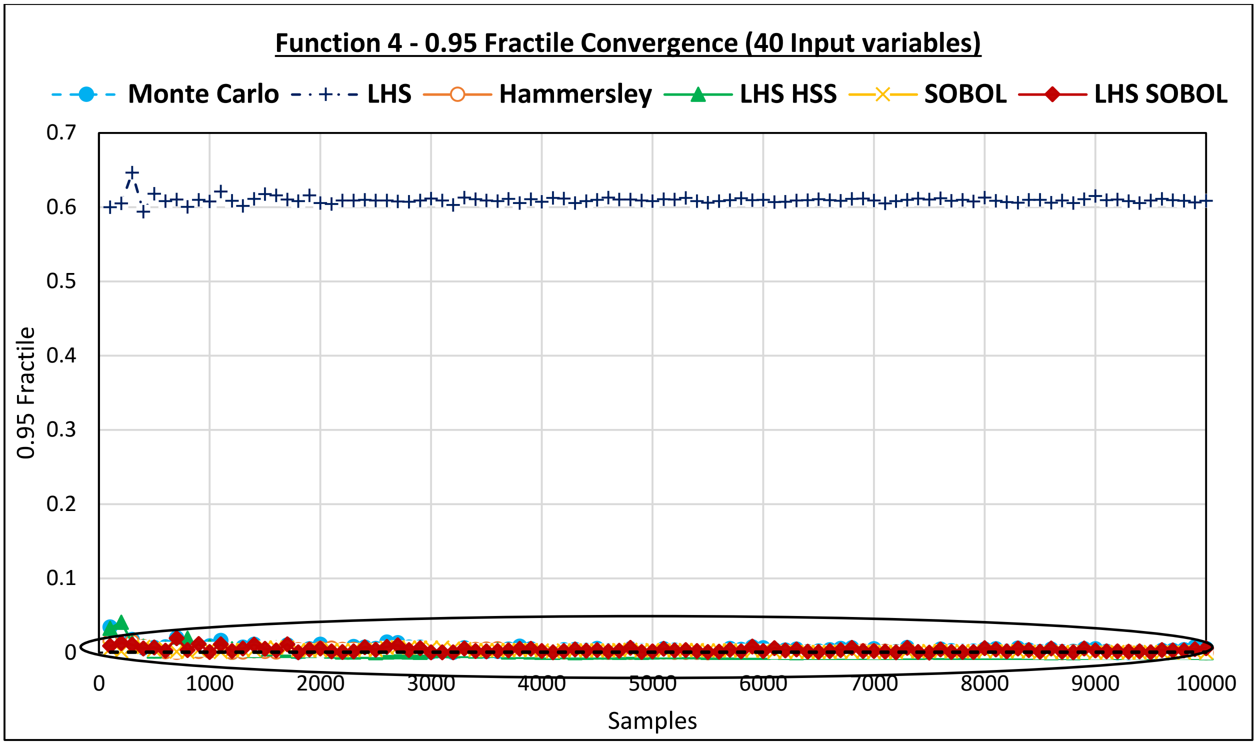

Figure 24. Fractile Vs. sample size for exponential function with 40 input variables for different sampling techniques

Zooming in

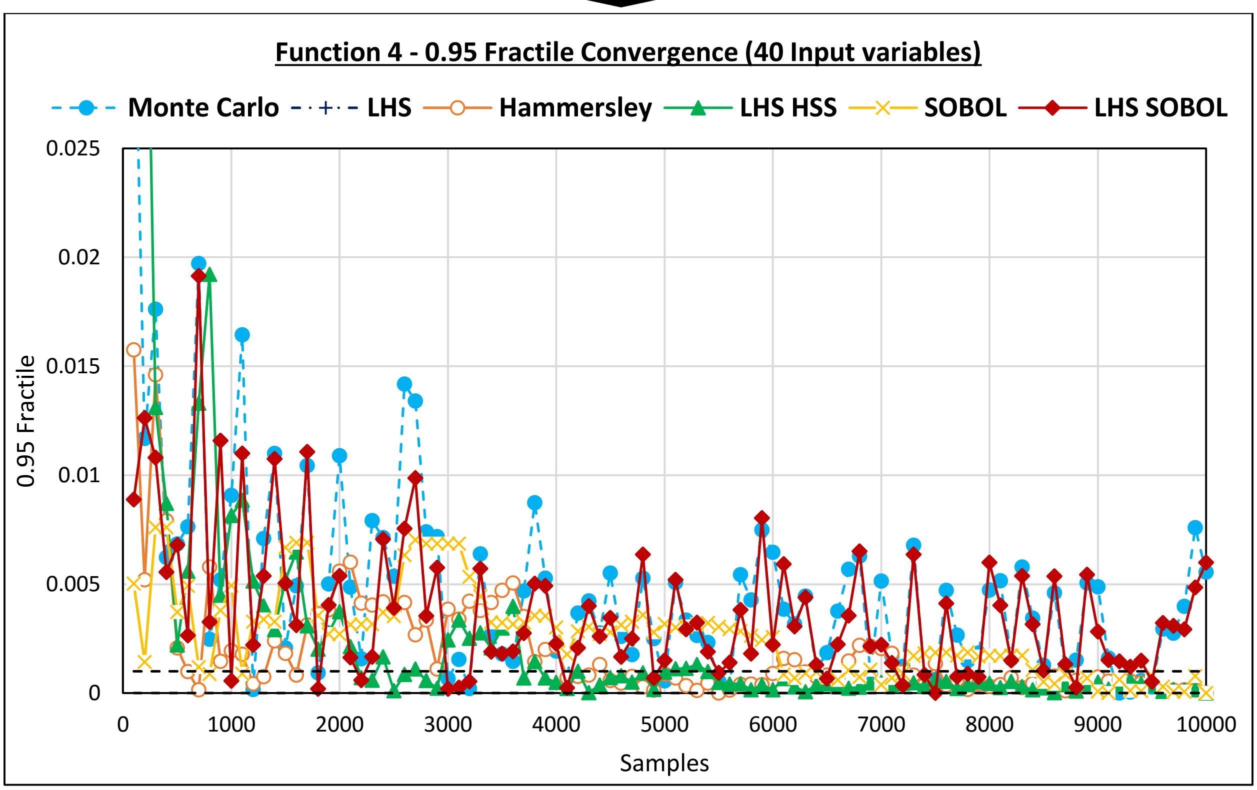

Figure 25. Zoomed in figure 24

Figure 22 to Figure 25 show results for 40 dimension problem for mean, variance and 0.95 fractile convergence respectively. Mean convergence results shown in Figure 22 show that MCS sampling technique does not converge within the error limits even for 10000 sample points. HSS and Sobol converge within the error limit with larger sample set but have lower efficiency as compared to LHS-HSS. LHS-Sobol converges within the error limit with less samples compared to HSS and Sobol, but LHS-HSS performs even better than LHS-Sobol.

Variance convergence results (Figure 23) show that LHS and MCS have not converged even with 10000 samples. HSS converges within the error limit with lesser number of samples compared to Sobol. LHS-Sobol eventually converge but it requires more samples compared to HSS, Sobol and LHS-Sobol.

Figure 24 shows 0.95 fractile results. From Figure 23, it is evident that LHS again has converged to different values which lies outside the error limits. From Figure 25, MCS and LHS-Sobol are fluctuating outside the error limits even with as high as 10000 samples. Sobol converges within the error limit but is has poor efficiency compared to HSS and LHS-HSS which has fastest convergence.

Looking at 40 dimension problem and mean, variance and 0.95 fractile convergence collectively, LHS-HSS shows consistent good results.

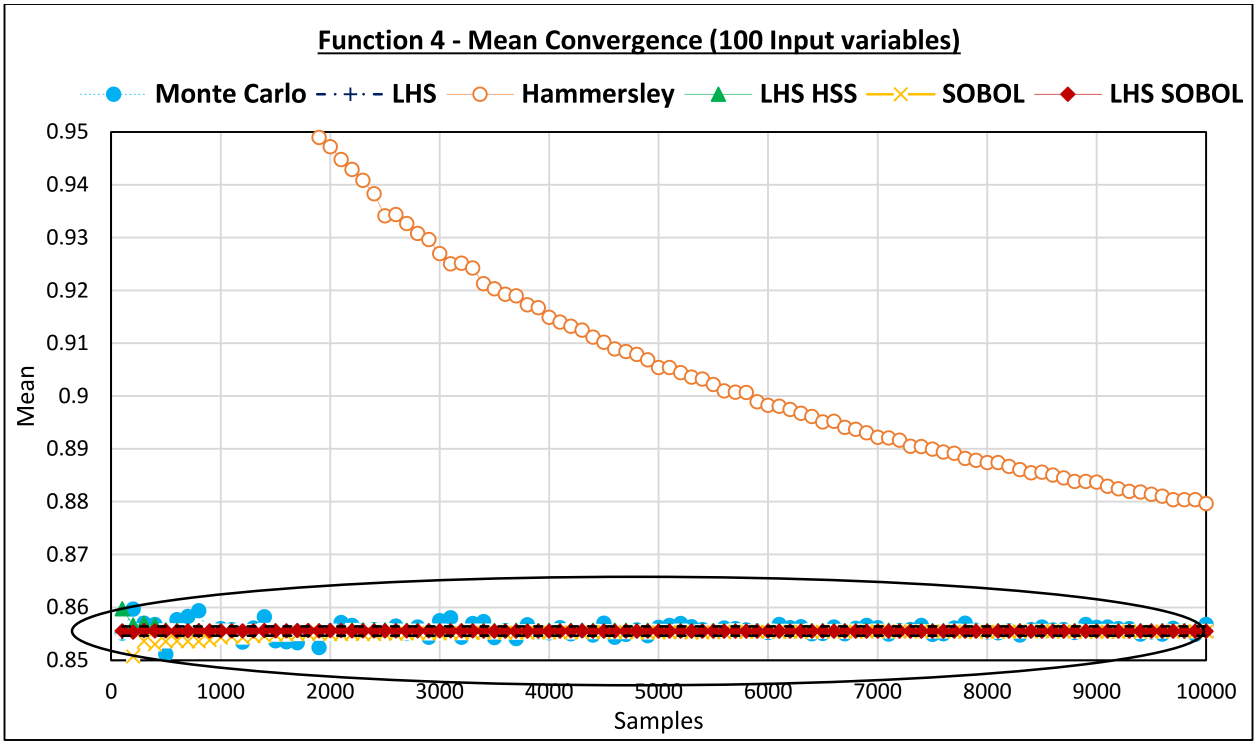

Figure 26. Mean Vs. sample size for exponential function with 100 input variables for different sampling techniques

Zooming in

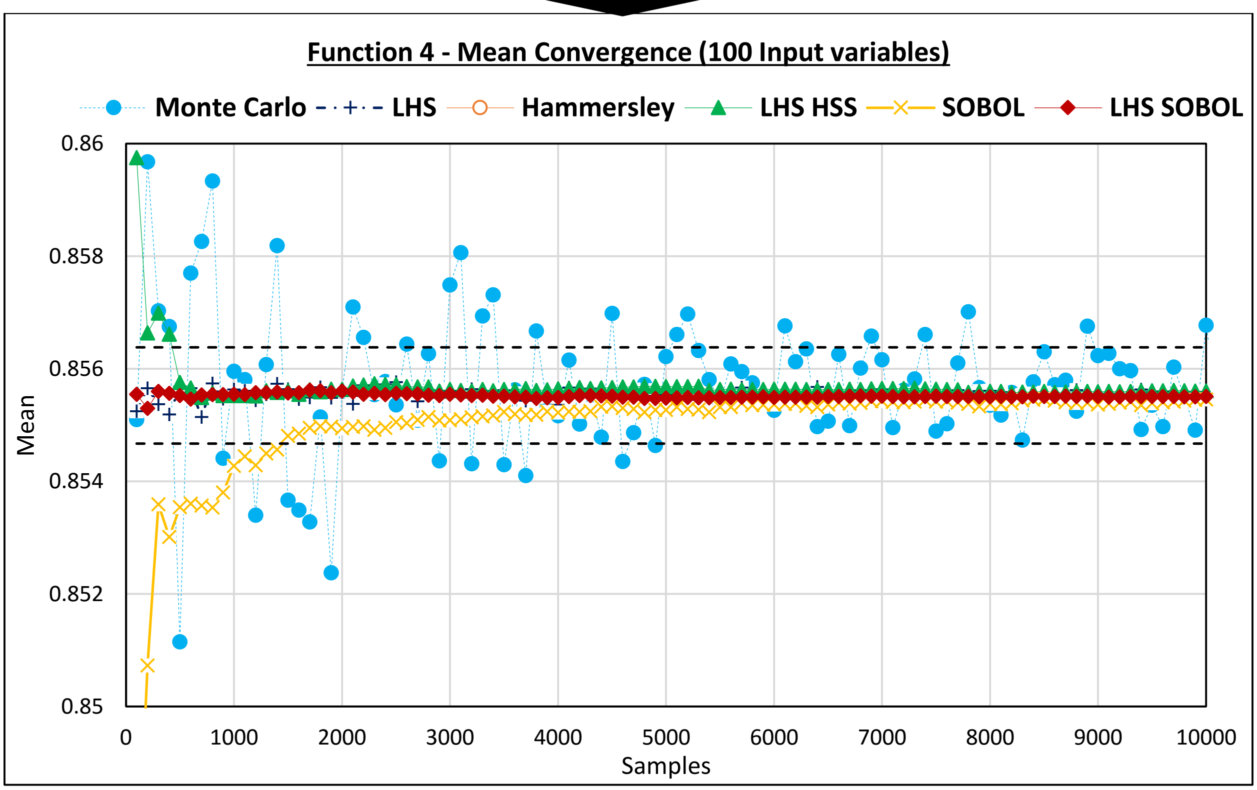

Figure 27. Zoomed in figure 26

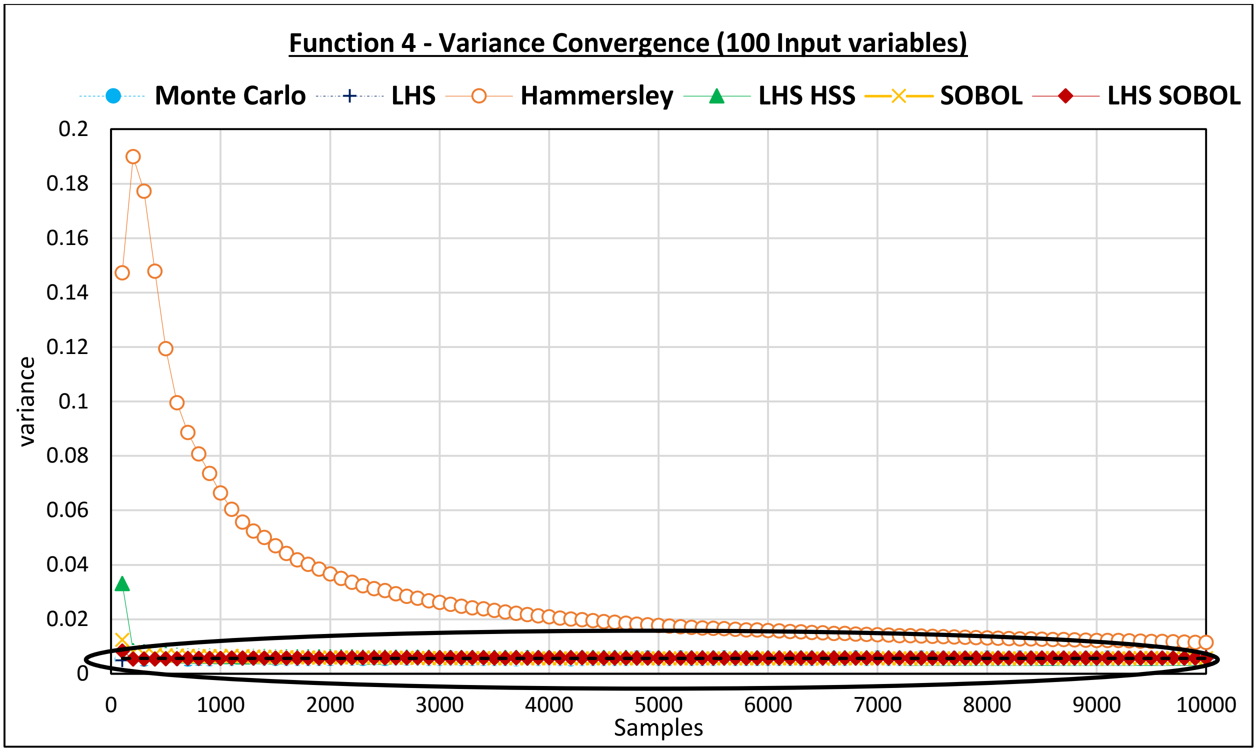

Figure 28. Variance Vs. sample size for exponential function with 100 input variables for different sampling techniques

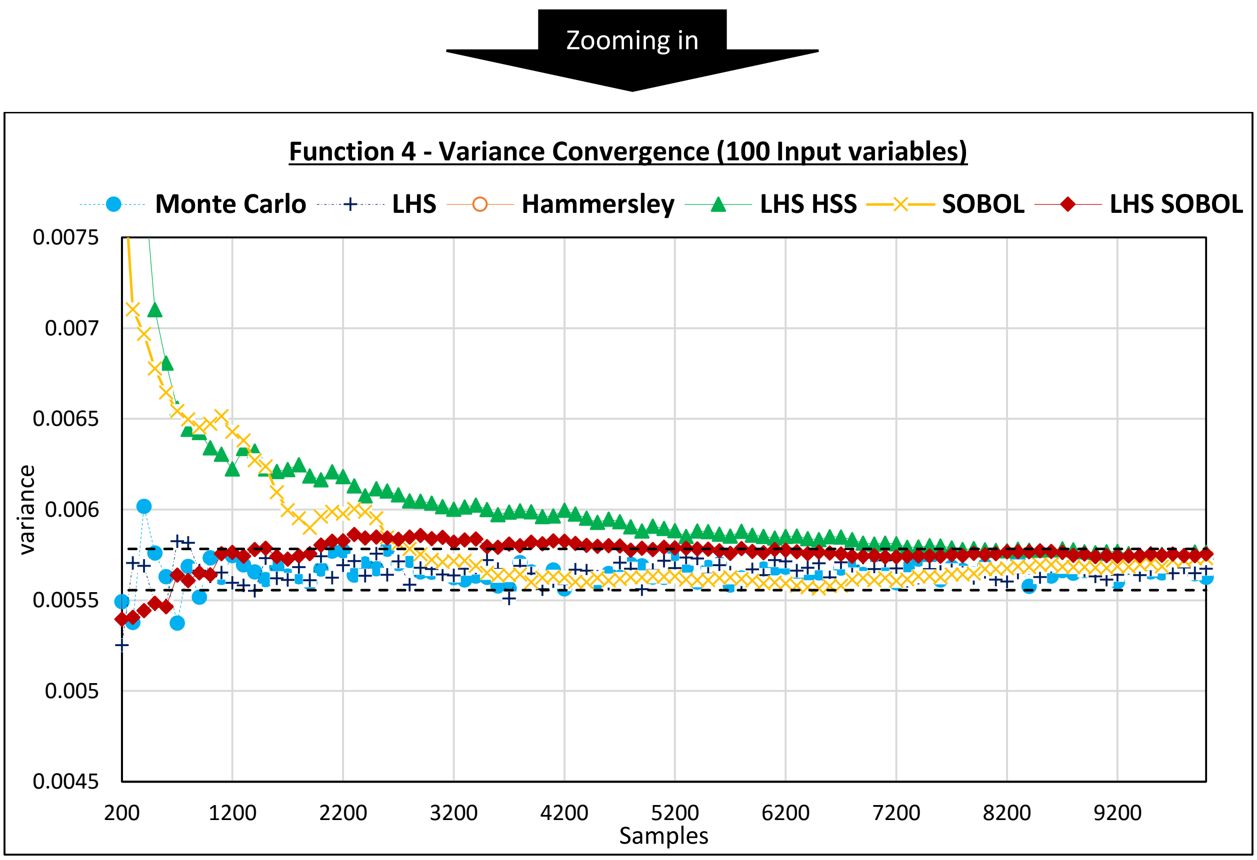

Figure 29. Zoomed in figure 28

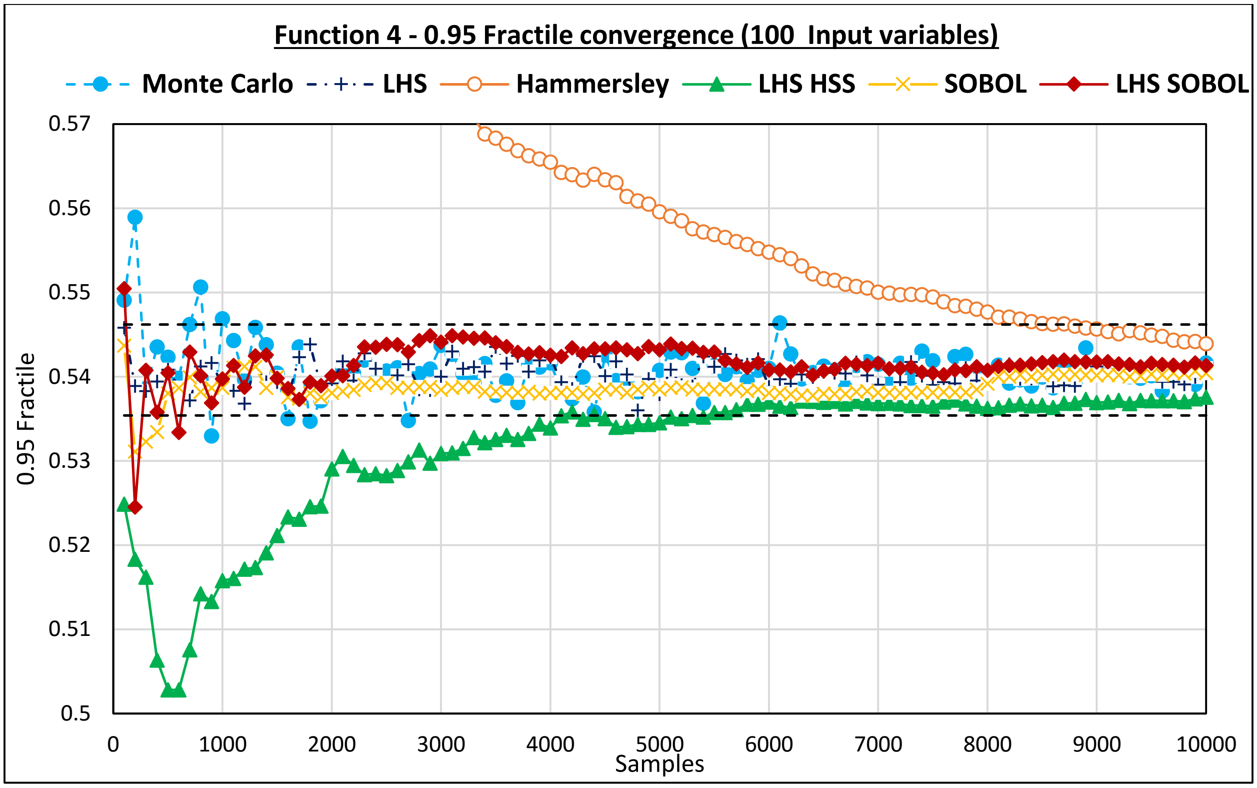

Figure 30. Fractile Vs. sample size for exponential function with 100 input variables for different sampling techniques

Figure 26 to Figure 30 shows results for 100 dimension problem for mean, variance and 0.95 fractile respectively. From mean convergence results shown in Figure 26, it is visible that HSS failed to converge within the error limits even for 10000 samples. From Figure 27, it is visible that MCS did not converge also. LHS-HSS and Sobol converge within the error limits, but require more number of samples compared to LHS-Sobol. Variance convergence results shown in Figure 28 show that again HSS failed to converge. Figure 29 shows that MCS, LHS and Sobol converge within the error limits with less number of samples compared to LHS-HSS and LHS-Sobol. 0.95 fractile convergence results in Figure 30, show that HSS failed to converge. MCS is also not converged. LHS-HSS converge within the error limits but with relatively larger number of samples compared to LHS, Sobol sand LHS-Sobol techniques. This shows that LHS-SOBOL performs best for all three evaluations for 100 uncertain variables. Sobol also did well.

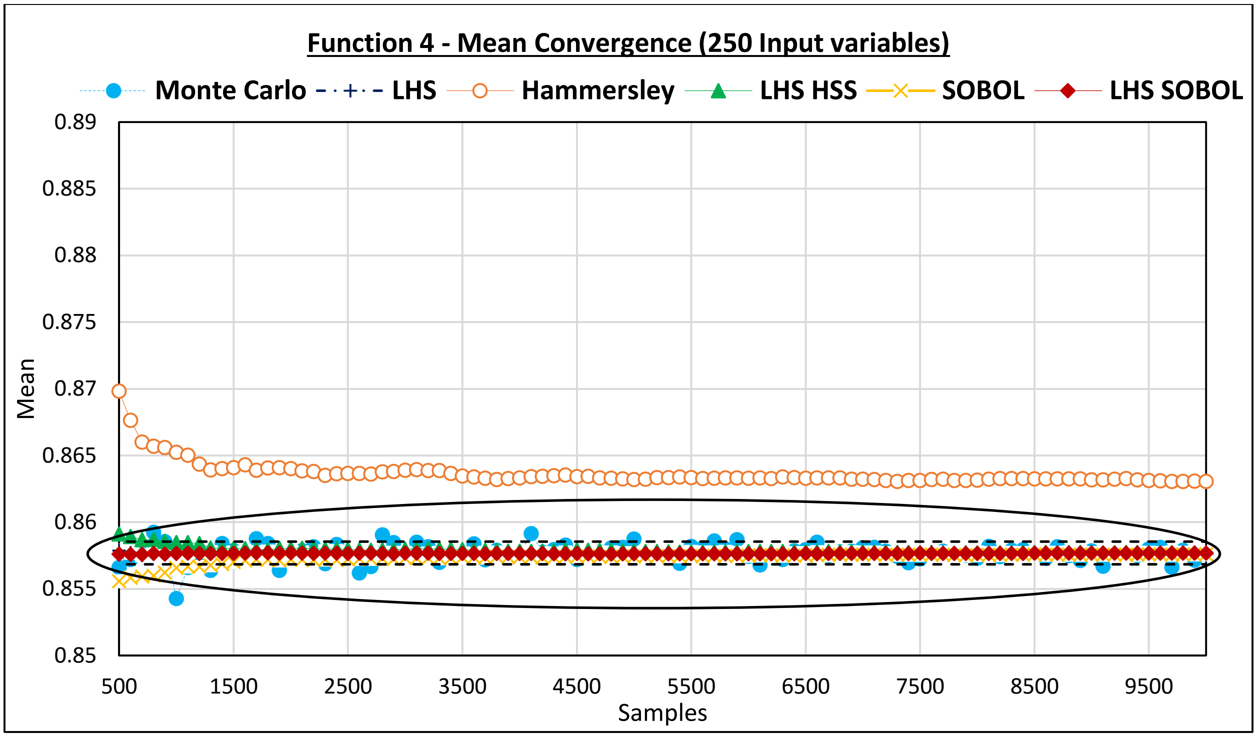

Figure 31. Mean Vs. sample size for exponential function with 250 input variables for different sampling techniques

Zooming in

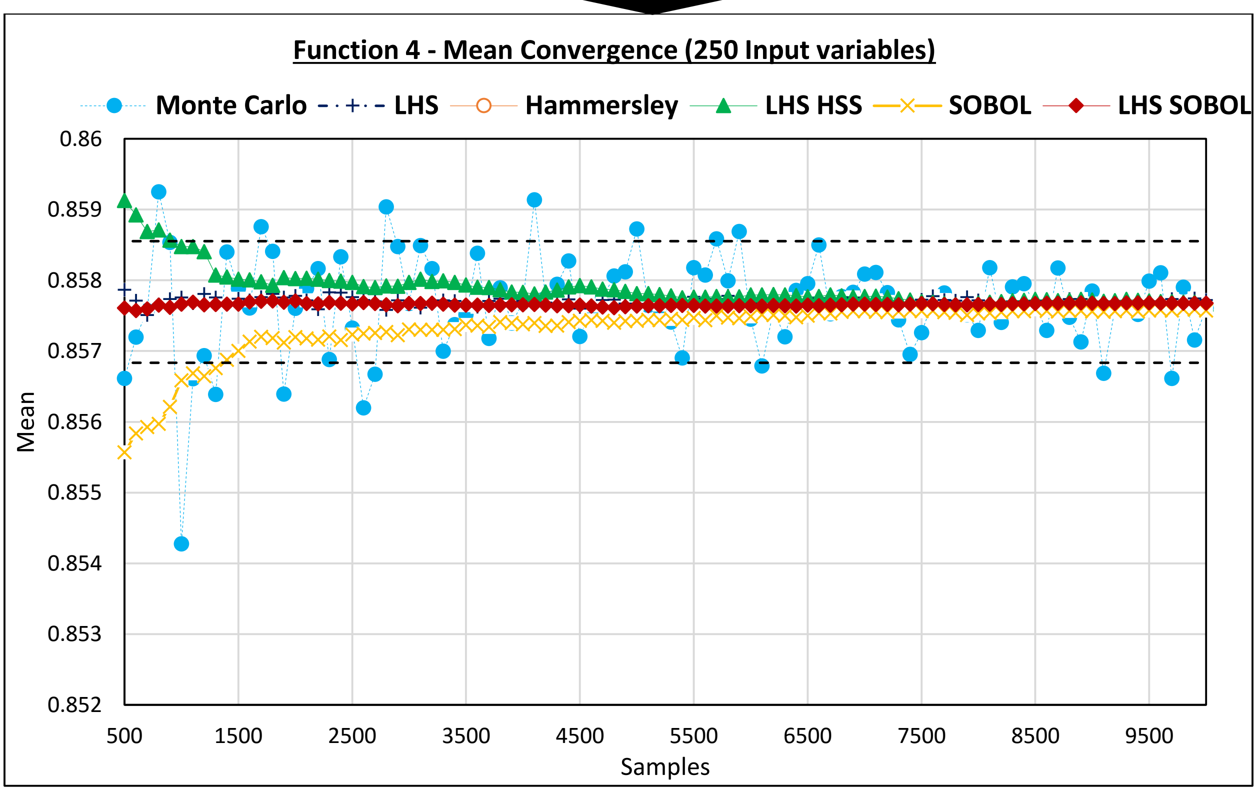

Figure 32. Zoomed in figure 31

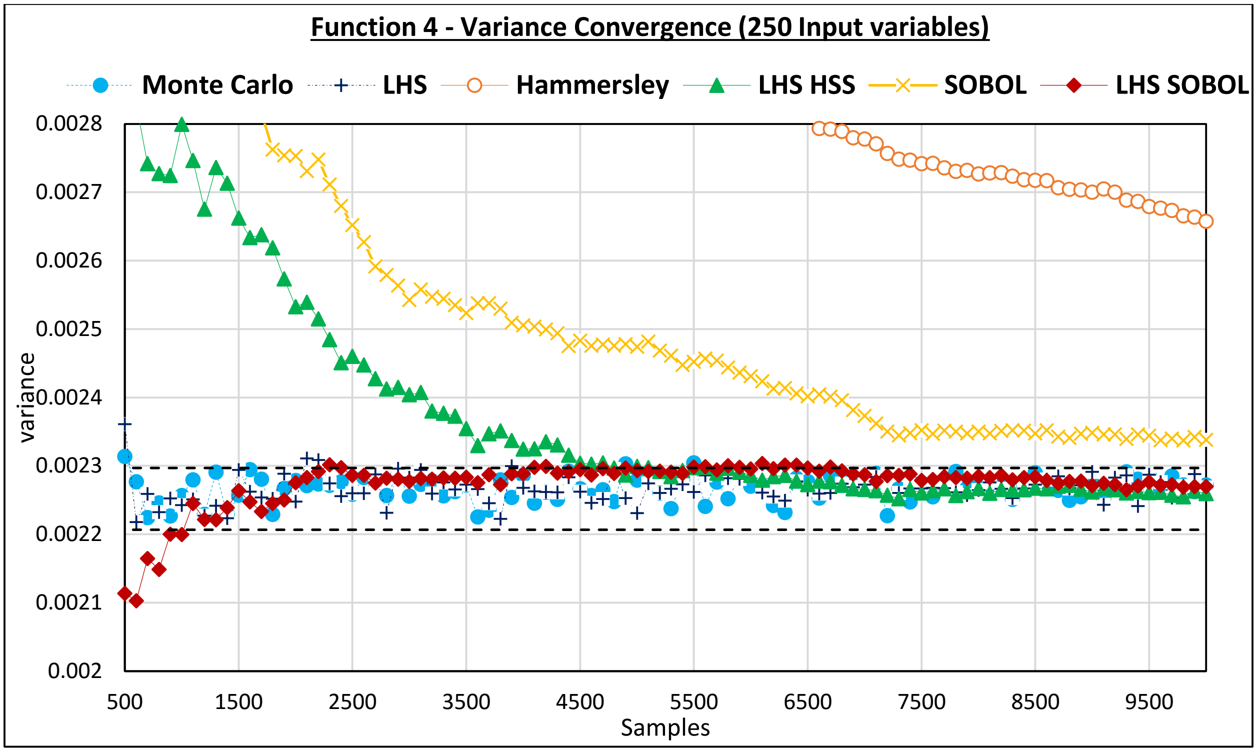

Figure 33. Variance Vs. sample size for exponential function with 250 input variables for different sampling techniques

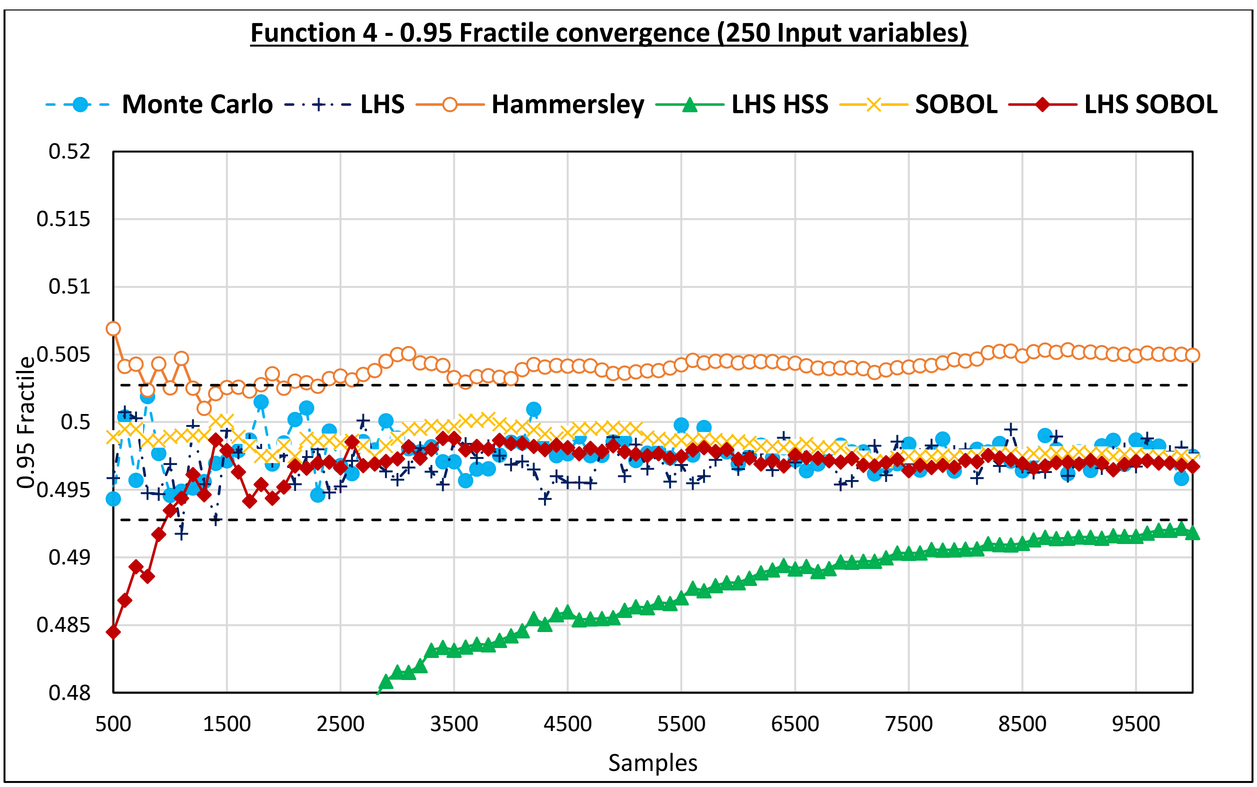

Figure 34. Fractile Vs. sample size for exponential function with 250 input variables for different sampling techniques

Figure 31 to Figure 34 show the results for mean, variance and 0.95 fractile for 250 dimensional problem respectively. We can see that the three top sampling techniques for 250 dimensions are SOBOL, LHS, and LHS-SOBOL. LHS-SOBOL is the best of all the techniques.

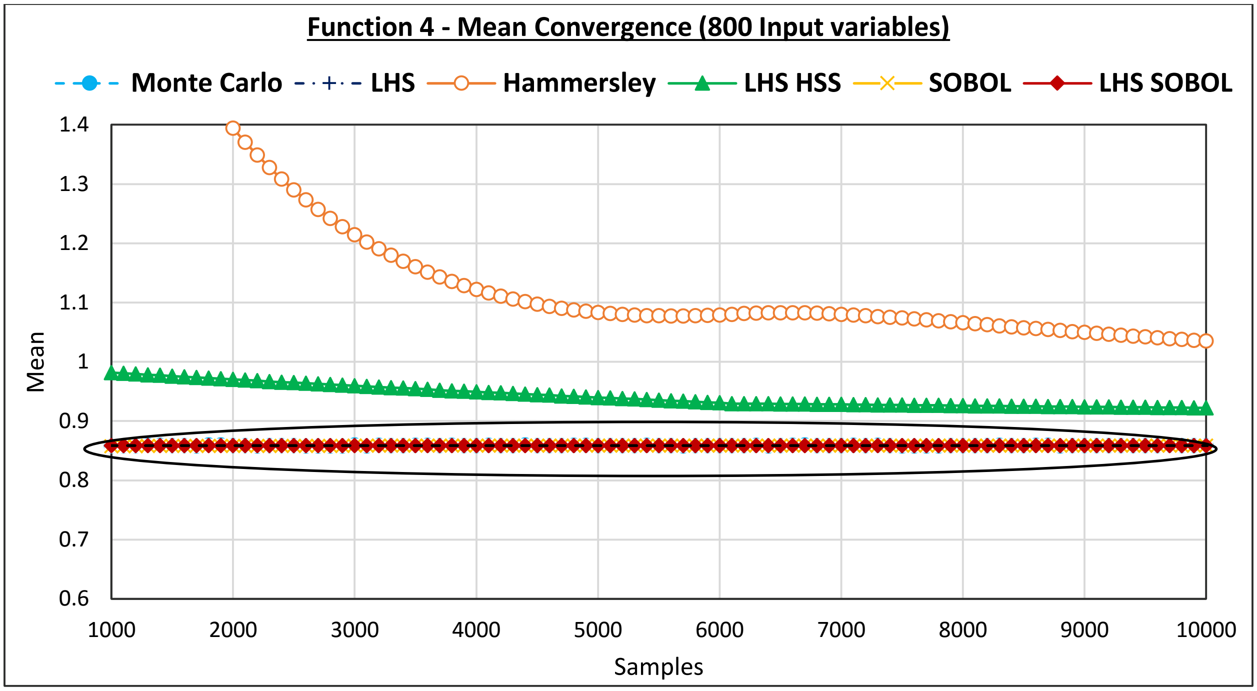

Figure 35. Mean Vs. sample size for exponential function with 800 input variables for different sampling techniques

Zooming in

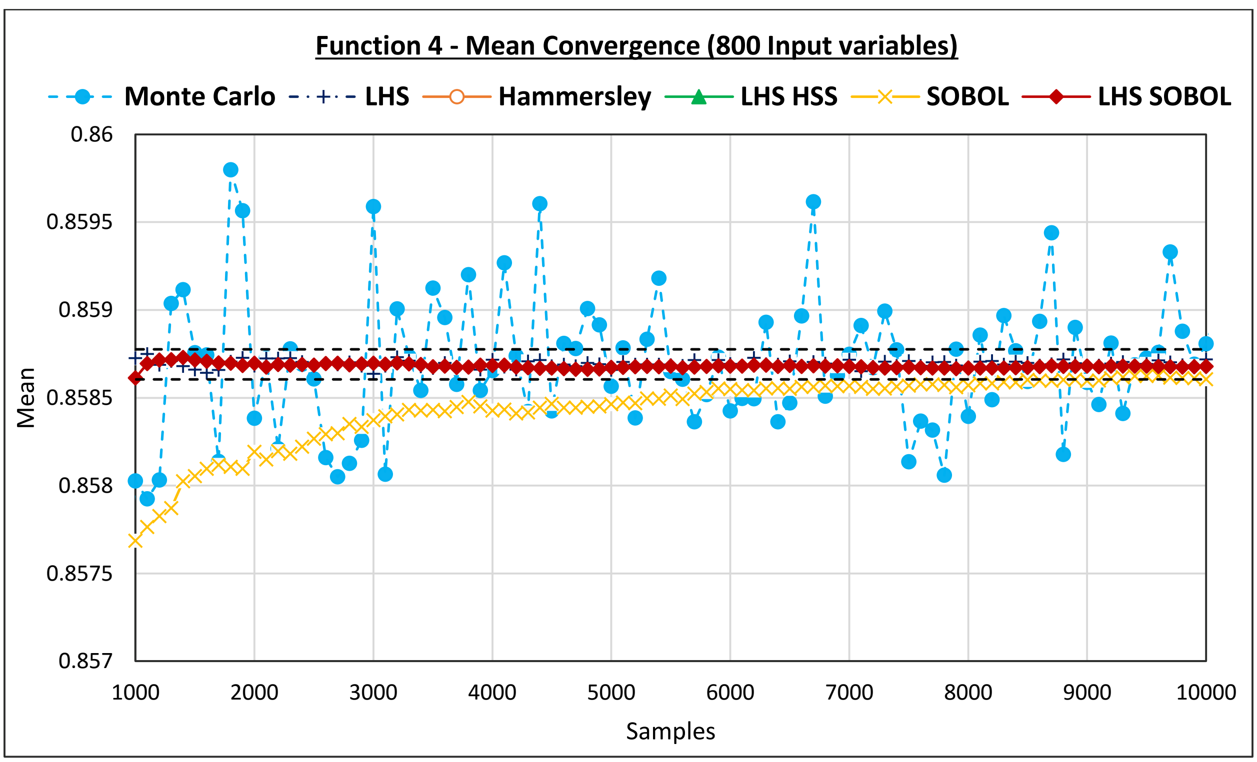

Figure 36. Zoomed in figure 35

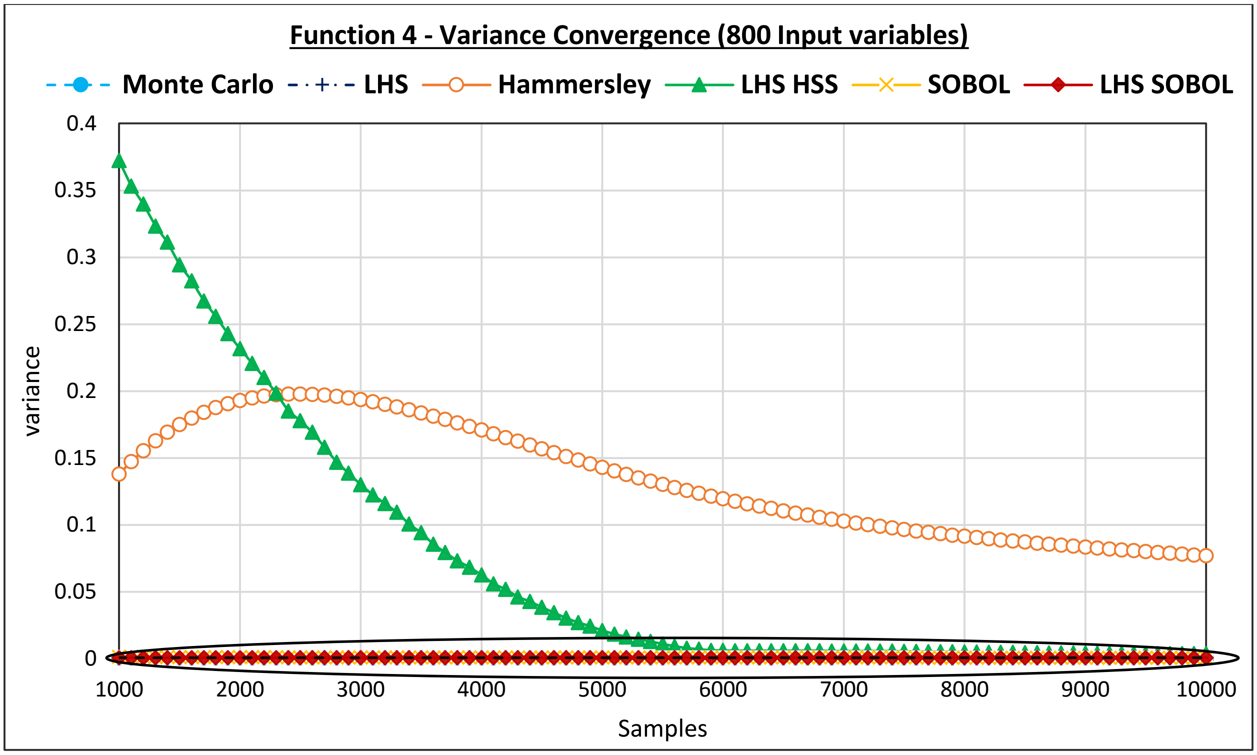

Figure 37. Variance Vs. sample size for exponential function with 800 input variables for different sampling techniques

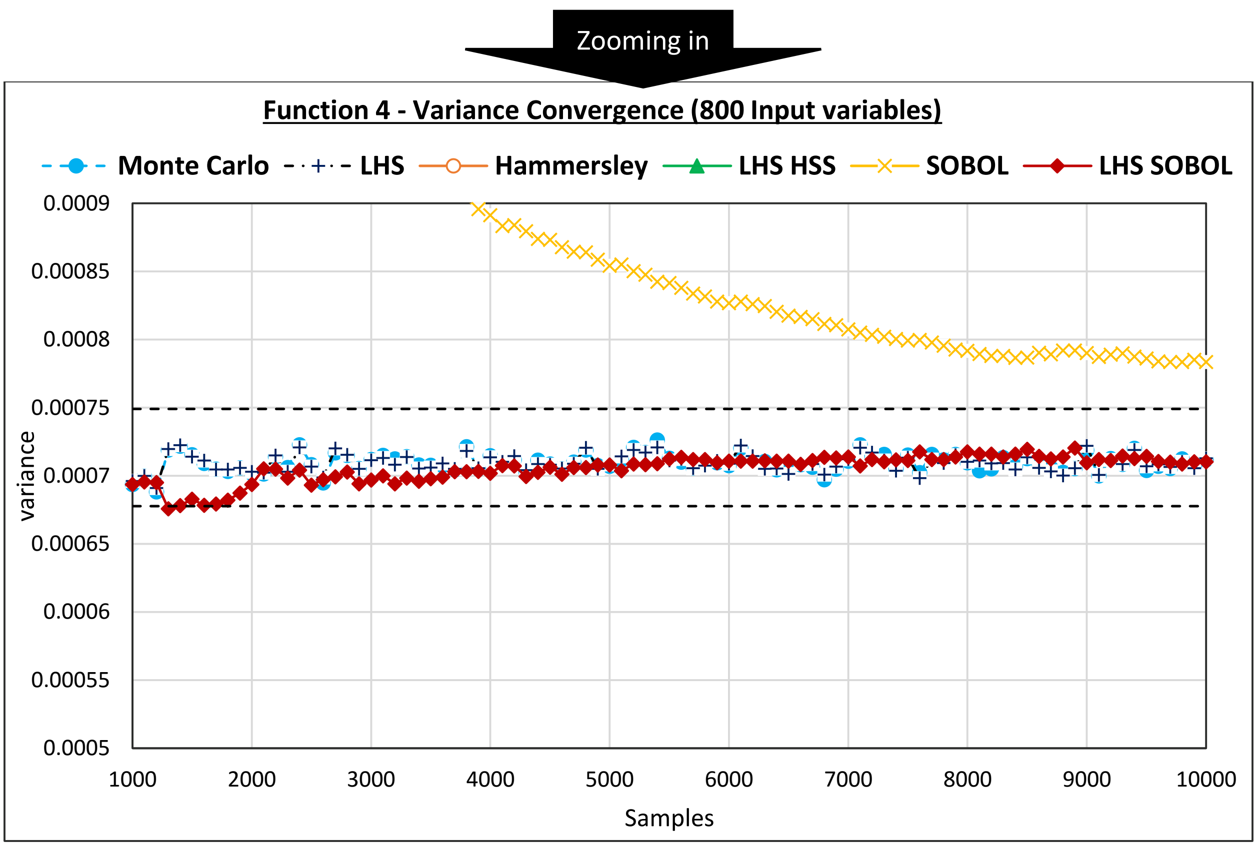

Figure 38. Zoomed in figure 37

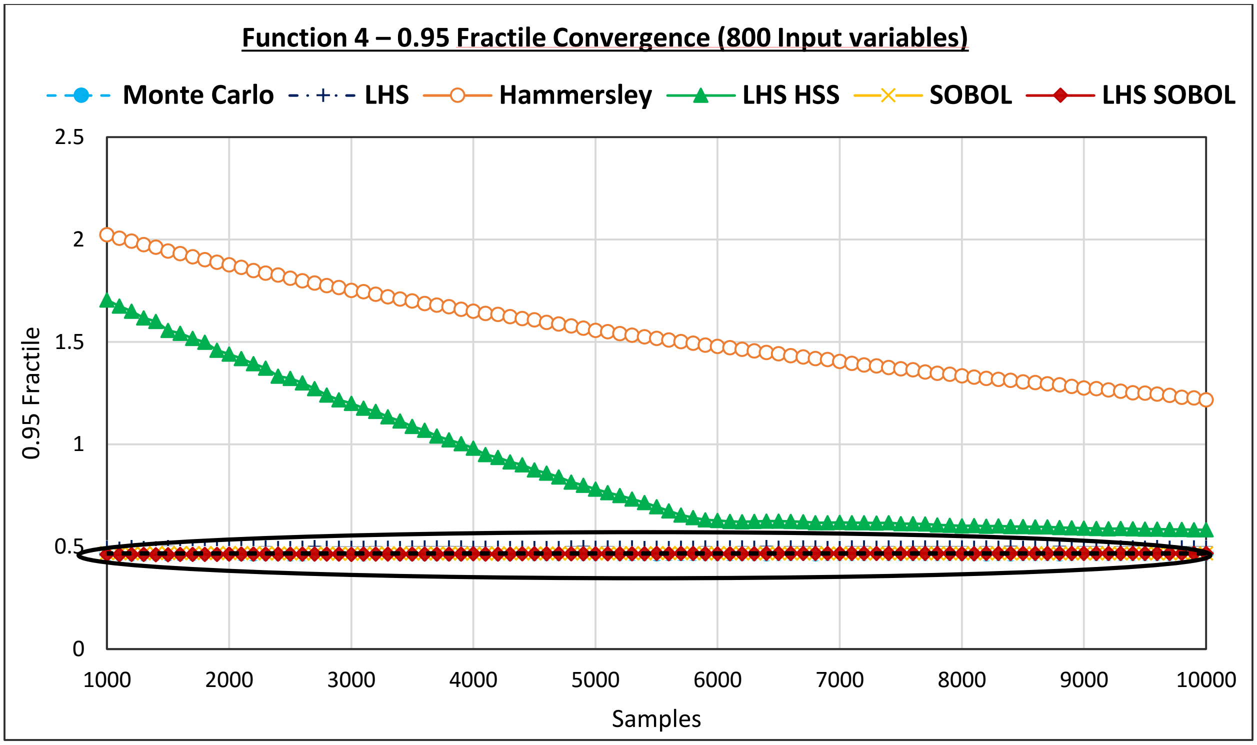

Figure 39. Fractile Vs. sample size for exponential function with 800 input variables for different sampling techniques

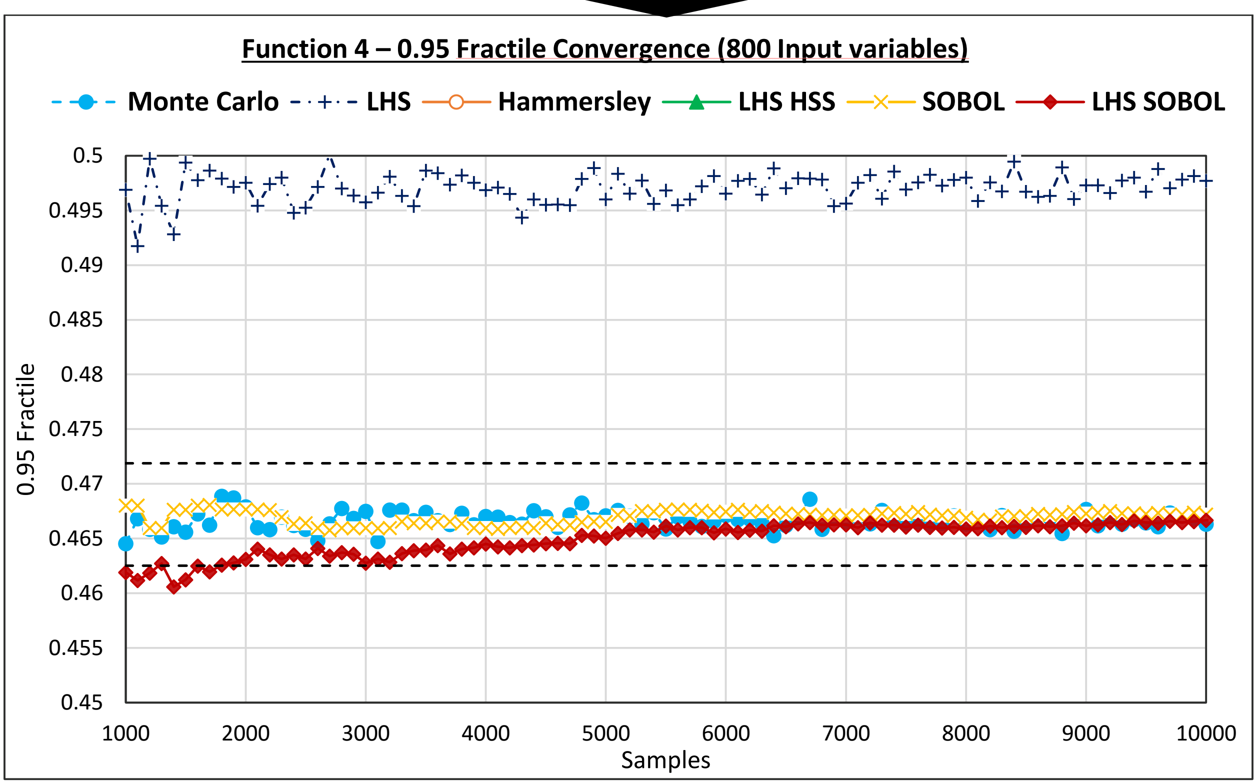

Figure 40. Zoomed in figure 39

Figure 35 to Figure 39 provide the results for mean, variance and 0.95 fractile for 800 dimensional problem respectively. From the results of mean shown in Figure 35, it is clearly evident that both HSS and LHS-HSS fail to converge within the error limits even for 10000 samples. From Figure 36 show that MCS also need large number of samples to converge. LHS shows as good results as LHS-Sobol for mean convergence. Sobol converges within the error limit with but it requires more number of samples compared to LHS-Sobol. From Figure 37 for variance, HSS and LHS-HSS fail to converge. LHS and LHS-SOBOL shows promising results. However, looking at the fractile results shown in Figure 39, LHS converges to wrong value but LHS-SOBOL and SOBOL perform the best. Therefore, considering all three mean, variance, and fractile evaluations for the 800 dimensional problem LHS-SOBOL performs the best.

The graphs shown so far showed only one function, in order to consider all test functions we plotted results of convergence for various test functions in Figure 41 to Figure 49. The convergence criterion used is number of samples required to converge within the error limit of the true value. The x-axis denotes the dimension of the problem and y axis the convergence criteria using various sampling technique. Figure 41, Figure 42 and Figure 43 show that for function 1, HSS and LHS-HSS show better performance than other sampling techniques up to 40 dimensions, LHS-HSS being the best. SOBOL, LHS-HSS, and LHS-SOBOL are better for 100 dimensions, LHS-SOBOL being the best at and beyond 100 dimensions. Similar results are seen for other functions (Figure 44-Figure 49).

In summary, HSS and LHS-HSS are good up to 40 uncertain variables but LHS-HSS outperforms HSS. SOBOL, LHS-HSS, and LHS-SOBOL are better from 40 to 100 uncertain variables and LHS-SOBOL is best after 100 uncertain variables.

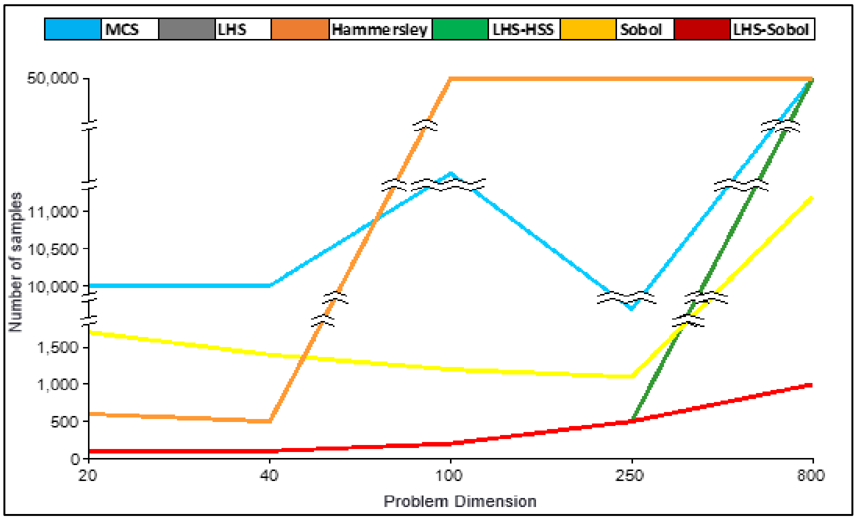

Figure 41. Function 1 mean convergence results

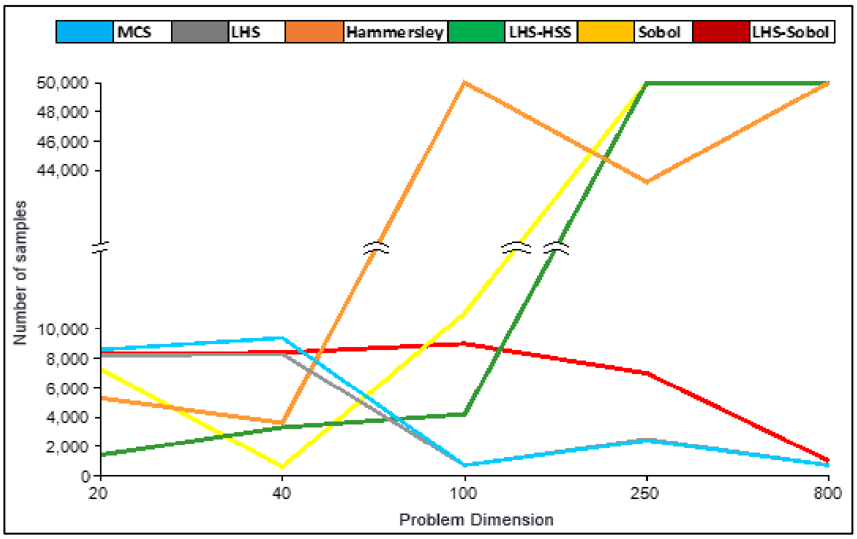

Figure 42. Function 1 variance convergence results

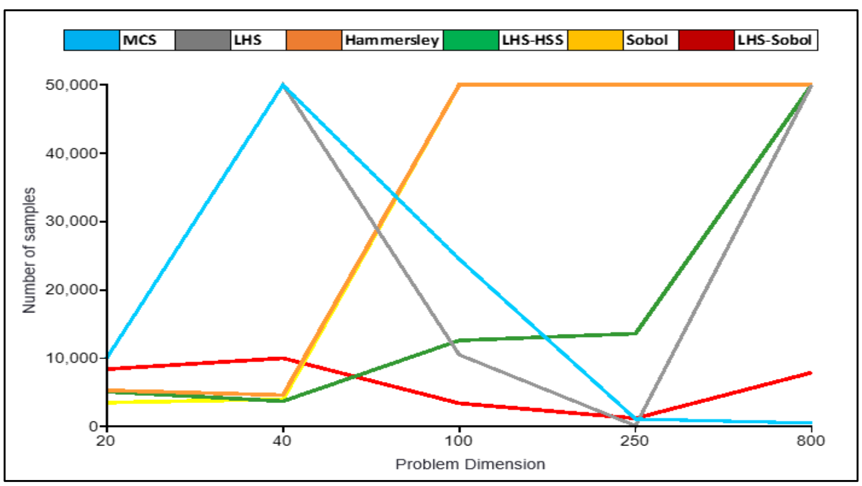

Figure 43. Function 1 - 0.95 fractile convergence results

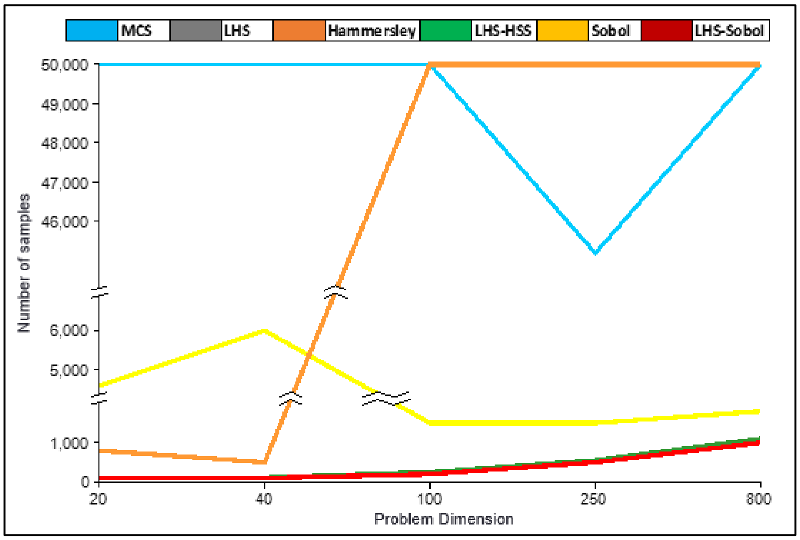

Figure 44. Function 3 mean convergence results

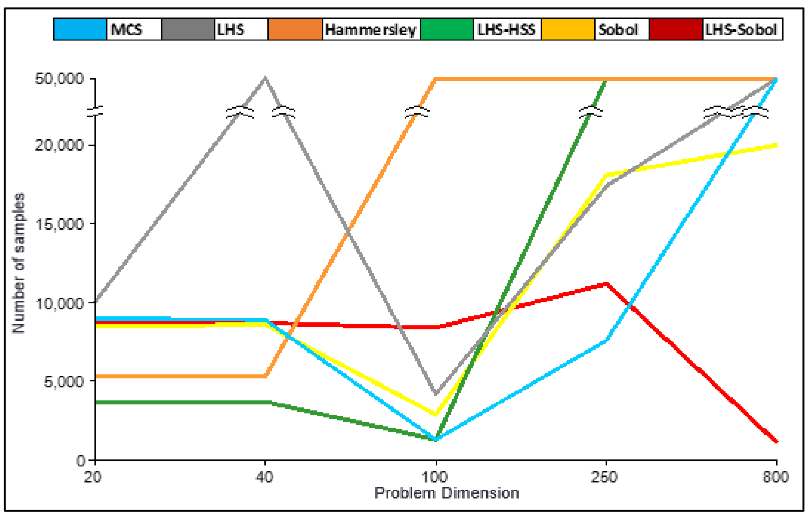

Figure 45. Function 3 variance convergence results

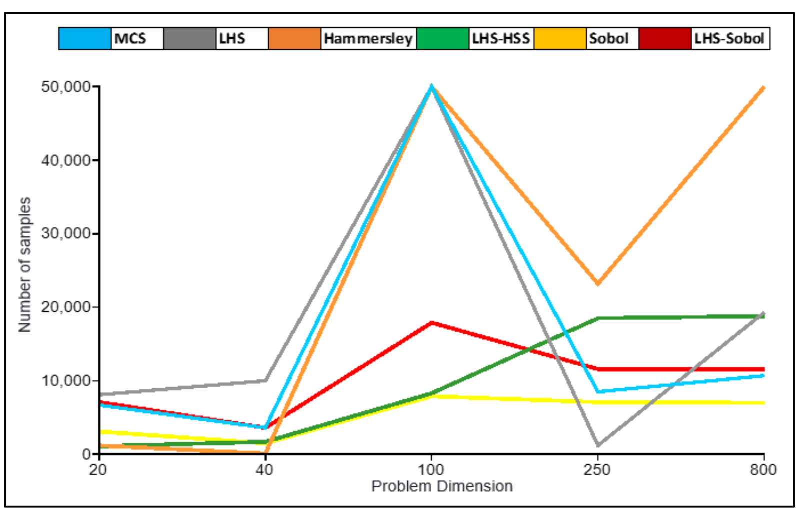

Figure 46. Function 3 - 0.95 fractile convergence results

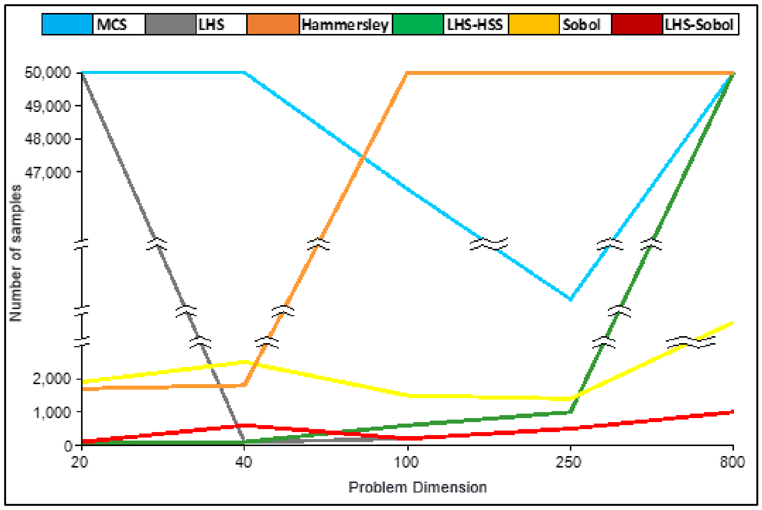

Figure 47. Function 4 mean convergence results

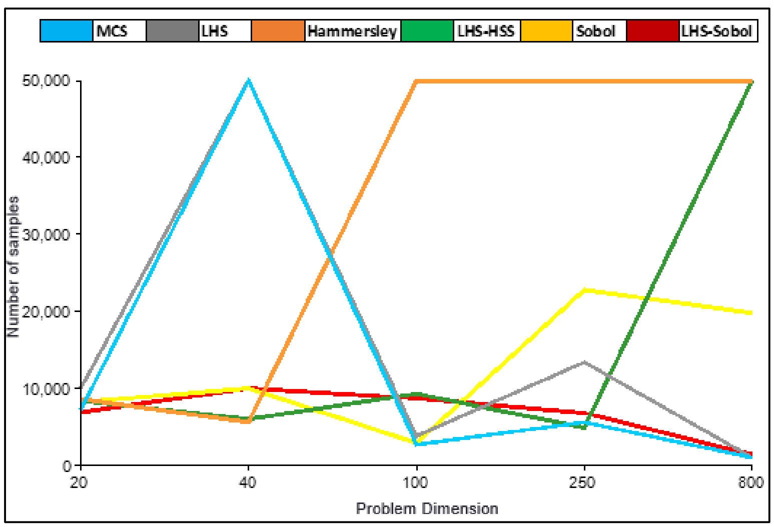

Figure 48. Function 4 variance convergence results

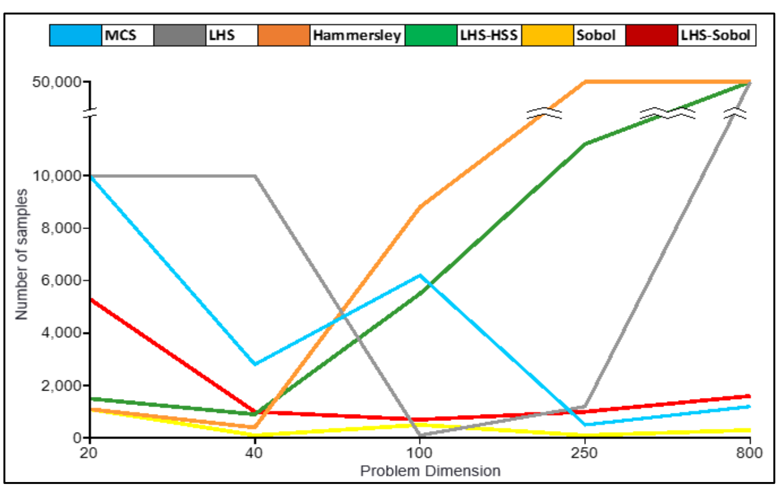

Figure 49. Function 4 - 0.95 fractile convergence results

Chapter 5. SUMMARY & FUTURE WORK

Real world stochastic optimization/ programming problems involve large scale uncertainties. This thesis presents a novel sampling technique for large scale uncertainties. It also analyzed various sampling techniques by conducting experiments with small to large dimensional functions.

5. 1. Summary

It has been found that Monte Carlo sampling technique is not efficient sampling technique because it lacks the uniformity property to cover the k-dimensional distribution. Latin Hypercube Sampling based on division of each variable distribution into equi-probable strata is one dimensionally uniform and performs better than Monte Carlo. Quasi Monte Carlo techniques like Hammersley and Sobol Sampling are multi-dimensionally uniform but lack one-dimensional uniformity like LHS. New sampling techniques like LHS-HSS and the proposed sampling technique LHS-SOBOL maintain uniformity in one-dimension as well as multi-dimension. These six sampling techniques are analyzed in this thesis using 216000 experiments using various functions. Convergence of mean, variance, and 95-th percentile (0.95 fractile) are used as criteria for measurement of efficiency of these sampling techniques. It has been found that HSS provides good convergence up to 40 uncertain variables. Sobol provides good convergence above 40 up to 100 variables and LHS-HSS up to 250 variables. LHS-Sobol provides good convergence above 40 up to 800 variables. We recommend that LHS-HSS can be used for problems up to 100 variables and LHS-Sobol should be used for problems beyond 100 variables.

5. 2. Future Work

In this work, we studied the sampling techniques up to 800 parameters. In this future, we plan to do more experiments with higher dimensions to see the applicability of this novel sampling technique.

Large scale uncertainties abound in financial literature. The new sampling technique proposed in this work will be very useful for solving problems in financial literature. Therefore, in the future we will try to compare this new sampling with existing methods in finance.

The large scale supply chain problem presented in the chapter 1 was solved using Chance constrained programming. We plan to solve the same problem using the generalized stochastic optimization framework (Figure 1) presented in Chapter 1 with our new proposed sampling technique. It is expected that this generalized approach will provide better results.

REFERENCES

1. Burhenne, Sebastian, Dirk Jacob, and Gregor P. Henze. 2011. "Sampling Based on Sobol'sequences for Monte Carlo Techniques Applied to Building Simulations." In *Proc. Int. Conf. Build. Simulat*, 1816–23. https://www.ibpsa.org/proceedings/BS2011/P_1590.pdf.
2. Caflisch, Russel E. 1998. "Monte Carlo and Quasi-Monte Carlo Methods." *Acta Numerica* 7: 1–49.
3. Cheng, Jian, and Marek J. Druzdzel. 2000. "Computational Investigation of Low-Discrepancy Sequences in Simulation Algorithms for Bayesian Networks." In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 72–81. Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm?id=2073956.
4. Chi, Hongmei, Peter Beerli, Deidre Evans, and Micheal Mascagni. 2005. "On the Scrambled Sobol Sequence." In , 775–82. Springer.
5. Dalal, Ishaan L., Deian Stefan, and Jared Harwayne-Gidansky. 2008. "Low Discrepancy Sequences for Monte Carlo Simulations on Reconfigurable Platforms." In *Application-Specific Systems, Architectures and Processors, 2008. ASAP 2008. International Conference on*, 108–13. IEEE. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4580163.
6. Diwekar, Urmila M. 2008. "Sampling Chapter." In *Introduction to Applied Optimization*, Second Edition. Vol. Volume 22. Springer.
7. Diwekar, Urmila M., and Jayant R. Kalagnanam. 1997. "Efficient Sampling Technique for Optimization under Uncertainty." *AIChE Journal* 43 (2): 440–47.
8. Diwekar, Urmila, and Mitchell J. Small. 2002. "11. Process Analysis Approach to Industrial Ecology." *A Handbook of Industrial Ecology*, 114.
9. Geoffrion, A. M. and Graves, G. W.(1974), Multicommodity distribution system design by benders decomposition. Management science, 20(5):822-844.

10. Helton, J.C., and F.J. Davis. 2003. “Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems.” *Reliability Engineering & System Safety* 81 (1): 23–69. doi:10.1016/S0951-8320(03)00058-9.

11. Iman, R.J. and Shortencarier, M.J.(1984) A FORTRAN77 Program and User's Guide for Generation of Latin Hypercube and Random Samples for Use with Computer Models, *NUREG/CR-3624, SAND83-2365,* Sandia National Laboratories, Albuquerque, N.M.

12. Joe, Stephen, and Frances Y. Kuo. 2008. “Constructing Sobol Sequences with Better Two-Dimensional Projections.” *SIAM Journal on Scientific Computing* 30 (5): 2635–54. doi:10.1137/070709359.

13. Joe, Stephen, and Frances Y. Kuo. 2008a. “Notes on Generating Sobol Sequences.” http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.626.8819&rep=rep1&type=pdf.

14. Joy, Corwin, Phelim P. Boyle, and Ken Seng Tan. 1996. “Quasi-Monte Carlo Methods in Numerical Finance.” *Management Science* 42 (6): 926–38.

15. Kalagnanam J. R. and U. M. Diwekar (1995), An efficient sampling technique for off-line quality control, *to appear in Technometrics.*

16. Klose, A. and Drexl, A. (2005), Facility location models for distribution system design. European Journal of Operational Research, 162(1):4-29.

17. Knuth, D.E. (1973), The Art of Computer Programming, Volume 1: Fundamental Algorithms, Addison-Wesley Publishing Co., Reading MA.

18. Kocis, Ladislav, and William J. Whiten. 1997. “Computational Investigations of Low-Discrepancy Sequences.” *ACM Transactions on Mathematical Software (TOMS)* 23 (2): 266–94.

19. Levy, George. 2002. “An Introduction to Quasi-Random Numbers.” *Numerical Algorithms Group Ltd., Http://www. Nag. Co. uk/IndustryArticles/introduction_to_quasi_random_numbers. Pdf (last Accessed in April 10, 2012)*, 143.

20. Lindemann, Stephen R., and Steven M. LaValle. 2003. “Incremental Low-Discrepancy Lattice Methods for Motion Planning.” In *Robotics and Automation, 2003. Proceedings. ICRA’03. IEEE International Conference on*, 3:2920–27. IEEE. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1242039.
21. Morgan G. and M. Henrion *(1990),Uncertainty: A guide to dealing with uncertainty in quantitative risk and policy analysis,* Cambridge University Press, New York.
22. Morokoff, William J., and Russel E. Caflisch. 1994. “Quasi-Random Sequences and Their Discrepancies.” *SIAM Journal on Scientific Computing* 15 (6): 1251–79.
23. Mula, J., Poler, R., Garcia-Sabater, J., and Lario, F. (2006), Models for production planning under uncertainty: A review. International journal of production economics, 103(1):271-285.
24. MURTHY, DN PRABHAKAR, and WALLACE R. BLISCHKE. 2004. “SAMPLING TECHNIQUES.” *Kirk-Othmer Encyclopedia of Chemical Technology*, 998.
25. Niederreiter, Harald. 1992. *Random Number Generation and Quasi-Monte Carlo Methods*. CBMS-NSF Regional Conference Series in Applied Mathematics 63. Philadelphia, Pa: Society for Industrial and Applied Mathematics.
26. Owen, Art B. 2003. “Quasi-Monte Carlo Sampling.” *Monte Carlo Ray Tracing: Siggraph* 1: 69–88.
27. Sahinidis, N. V. (2004), Optimization under uncertainty: state-of-the-art and opportunities. Computers & Chemical Engineering, 28(6):971-983.
28. Sobol, Ilya M., Danil Asotsky, Alexander Kreinin, and Sergei Kucherenko. 2011. “Construction and Comparison of High-Dimensional Sobol’Generators.” *Wilmott* 2011 (56): 64–79.
29. Thomas, D. J. and Gri_n, P. M.(1996), Coordinated supply chain management. European journal of operational research, 94(1):1-15.

30. Vidal, C. J. and Goetschalckx, M. (1997), Strategic production-distribution models: A critical review with emphasis on global supply chain models. European Journal of Operational Research, 98(1):1-18.

31. Wang, Renyou, Urmila Diwekar, and Catherine E. Grégoire Padró. 2004. “Efficient Sampling Techniques for Uncertainties in Risk Analysis.” *Environmental Progress* 23 (2): 141–57. doi:10.1002/ep.10020.

Nishant Dige

ndige2@uic.edu

MS Industrial Engineering *(GPA 3.85)*	- **University of Illinois at Chicago (UIC)**	- Aug 2014 - May 2016
BE Production Engineering *(GPA 3.4)*	- **University of Pune**	- Jun 2008 - May 2012

Master's Thesis

Effective sampling techniques for large scale stochastic optimization problem. Aug 2014 – May 2016

- Studied large scale supply chain optimization problem with 800 uncertainties.
- Developed Novel Sampling algorithm for stochastic optimization which reduces computational speed and intensity.
- Compared six different sampling techniques analyzing uniformity and efficiency on different dimensional problems.

Keywords: Quasi Monte Carlo Sampling, LHS-Sobol, Optimization under uncertainty, stochastic supply chain network problem

Publication:

Dige, Nishant & Diwekar, Urmila *"Effective sampling techniques for large scale stochastic optimization problem."*

INFORMS, Philadelphia, PA (2015, November - Conference Presentation)

Work Experience

CNH Industrials, New Holland, PA May 2015 - Aug 2015

Industrial Engineer Intern

- Implemented new shop floor layout considering operations sequence and logistics workflow to increase production rate.
- Re-designed and standardize the logistics delivery method using automated guided vehicles and created SOP document.
- Analyzed and improved human machine efficiency to 90% by eliminating non-value added activities & variation.
- Balanced line to match TAKT, reduced cycle time by *22%* and increased productivity gaining annual saving of *$400K*.
- Planned KAI to achieve and measured KPI (Cost/Benefit ratio) through each step within the project.

Tools: Lean Principles, Logistics Workflow, Automated Guided Vehicles, Kanban, Kaizen, Auto-Cad, Material Handling, Ergonomics

Larsen & Toubro, Technology Services, Baroda, India Jan 2013 - Jun 2014

Design Engineer

- Analyzed, optimized and designed components for cost reduction and manufacturability.
- Assisted manufacturing department in troubleshooting manufacturability issues & provided support in products release.
- Designed CAD models and manufacturing drawings for sheet metal components and assemblies of Harvesters & Baler's.

Tools: Pro-E, GD&T, FMEA, Cause and Effect, QFD, Concurrent and Reverse Engineering, Team Center, E-BOM, E-Change order

Alfa Laval, Pune, India Sep 2011 – Mar 2012

Industrial Engineering Intern

- Reduced lead time from Value Stream by identifying and eliminating non value added activities in material flow.
- Re-designed logistics supermarket layout & delivery system to increase productivity by *63%* delivering savings of *$2M*.

***Tools*:** Value Stream Mapping, logistics Replenishment, Line Balancing, Time Study, 5S, Spaghetti Diagram, SOP, Financial Metrics

Course Project

Supply chain retailing strategy considering fulfillment cost in Multi-Channel Retailing Jan 2015 – May 2015

- Developed transportation, labor & inventory holding cost models for analyzing demand scenarios for different channels.
- Analyzed lead time of supply chain & demand variation to generate safety inventory stock.
- Analyzed trade-offs for each channel in order to reduce fulfillment cost in meeting customer requirement.

***Tools*:** Retailing Strategy, SAP – SCM Aggregate Planning, Math program, Fulfillment Cost models, Excel Solver, Data Mining

Time-Series forecasting of the behavior of promoted trend Aug 2015 – Dec 2015

- Analyze trends in large scale stochastic data and regression estimation of next value on preceding values.
- Develop ARMA model for predicting highly complex data and data pattern analysis.

Tools: Time Series Forecasting, Trend and Seasonality Analysis, Joint Optimization, Large Stochastic Data, Matlab, Advanced Excel

Certification:

Lean Green Belt (IIE 2348-7120)

Tools Learned: JIT, Kanban, Takt time, Kaizen, 5 why's, 3M, 5S, Value stream mapping, u-Cell, Poka-yoke, PDCA, HERCA

Six Sigma Green Belt (IKSC Knowledge Bridge)

Tools Learned: DMAIC, CTQ analysis, Benchmarking, Pareto Chart, Fishbone Diagram, Statistical analysis, QFD, Control Charts