Evolutionary Programming Based Stratified Design Space Sampling

Brian K. Beachkofski *
Air Force Research Laboratory
1950 5th St., Building 18D
Wright–Patterson Air Force Base
Dayton, OH 45433
(brian.beachkofski@wpafb.af.mil)

Dr. Gary B. Lamont †
Air Force Institute of Technology
2950 P St., Bldg. 640 Room 212
Wright–Patterson Air Force Base
Dayton, OH 45433
(gary.lamont@afit.edu)

Abstract

Recently there have been advances in stratified sampling techniques that attempt to enforce equal distributions not only across the design variables, but also onto the design space itself. This requires a numerically intensive optimization routine. Until now, no optimization strategy was able to distribute sample points evenly in the design space, but Evolutionary Algorithms (EA) act as an enabling technology to spread in such a way that the minimum distance between points is near ideal for the design space. The proposed technique is applied to a standard probabilistic analysis problem, a one-degree of freedom oscillator simulating a blade near harmonic resonance. The Evolutionary Programming results compared to Latin Hypercube Sampling indicate a better estimate with a smaller confidence interval.

1 Introduction

A general trend in stratified sampling methods is to use more intelligence in selecting sample locations rather than resorting to brute force methods such as Monte Carlo (MC). This approach is seen in the development of Latin Hypercube Sampling (LHS) [1] and more recently Distributed Hypercube Sampling (DHS)[2]. This paper outlines another attempt to intelligently choose the sample points rather than leaving the process completely random.

The algorithm will then be applied to a standard probabilistic analysis problem in the turbine engine community. The problem is a Single Degree of Freedom (SDOF) Oscillator representing a turbine engine blade under harmonic resonance.

A review of several sampling methods puts the current work into perspective. The theory behind sampling method evaluation of the integration of the probability density function is also explained, with emphasis on how the proposed method leads to better validity of the equal area assumption.

The justification for employing Evolutionary Programming as opposed to other optimization routines follows the sampling method review. The overall flow of the algorithm is explained, as is each independent operator in the routine. Sufficient information to reproduce the results is presented as well.

The results of the algorithm are sets of sample points for sampling the response space. The quality measure is the confidence interval of the predicted probability of failure and the Coefficient of Variation of minimum distances between any two sample points. Finally, an Analysis of Variance is conducted to show that the differences between methods are likely real and not due to a random variation of points selected from the same distribution.

2 Problem Definition

A standard problem in probabilistic analysis for turbine engines is the single degree of freedom oscillator first defined by Griffiths et al[4], developed as part of their work on probabilistic turbine engine blade design. An elementary damped spring oscillator, Figure (1), under cyclic loading combines the stiffness, K, and the mass, M, into a single parameter, ω_n, the blade natural frequency at nominal rotational loading.

* Research Engineer, Propulsion Directorate, Turbine Engine Components Branch

† Professor of Computer Engineering, Department of Computer and Electrical Engineering

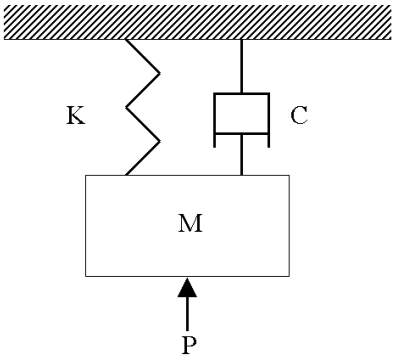

Figure 1: One-Degree of Freedom Oscillator.

The loading, P, is periodic with amplitude, P_0, and frequency of excitation, ω.

$$P(t) = P_0 \sin \omega t \tag{1}$$

The derivation of the limit state function that follows is the notation first used by Griffiths, but could be derived using any basic dynamics reference such as one by Rao [5]. An aerodynamic load generates the forcing function at an M per revolution disturbance typical of blade excitation due to inlet distortion or upstream (downstream) stators. With N as the rotor speed in revolutions per minute, the forcing frequency, in Hertz, can be written as:

$$\omega = \frac{M \cdot N}{60} \tag{2}$$

Finite Element Analysis (FEA) predicts the frequency of the blades under the centrifugal loading as a function of rotational speed. Using the slope of this function at the nominal speed, N_{nom}, a local approximation is made by a first order Taylor's Series expansion.

$$\omega_n = \omega_n(N_{nom}) + Slope(N - n_{nom}) \tag{3}$$

The critical damping ratio, ζ, is the damping value, C, divided by the value needed for critical damping, C_{crit}, and the frequency ratio, β, is the forcing frequency divided by the natural frequency. With this information, the steady state deflection amplitude is of the form:

$$\begin{aligned} \rho &= \frac{\frac{P_0}{K}}{\sqrt{(1 - (\frac{\omega}{\omega_n})^2)^2 + (2 \cdot \zeta \cdot \frac{\omega}{\omega_n})^2}} \\ &= \frac{\frac{P_0}{K}}{\sqrt{(1 - \beta^2)^2 + (2 \cdot \zeta \cdot \beta)^2}} \end{aligned} \tag{4}$$

Dividing by the numerator creates a non-dimensional dynamic amplification factor, D. The design limit of the blade is some percent, Y, of the nominal allowable steady state displacement.

$$\left(\frac{P_0}{K}\right)_{nom} \cdot D_{nom} = Y \cdot \rho_{allow_{nom}} \tag{5}$$

The variations of the forcing field as determined by Computational Fluid Dynamics change the load factor, LF, just as changes in the mode shape are determined by Finite Element Analysis are accounted for in the mode shape factor, MSF. These factors multiply the values of P_0 and K respectively:

$$\frac{P_0}{K} = \left(\frac{P_0}{K}\right)_{nom} \cdot LF \cdot MSF \tag{6}$$

The limit state function, ρ, is the difference between ρ and ρ allowable, or non-dimensionally as g:

$$\begin{aligned} Y \cdot LF \cdot MSF \cdot \frac{D(N, \omega_n, Slope, \zeta)}{D_{nom}} \cdot \frac{\rho_{allow_{nom}}}{\rho_{allow}} - 1 \\ = g(\omega_n, \zeta, \rho_{allow}, N, Slope, LF, MSF) \end{aligned} \tag{7}$$

When ρ exceeds ρ allowable, the function is greater than zero and the blade fails. Equation (7), the limit state function, is dependent on the seven variables in the problem: $\omega_n, \zeta, \rho_{allow}, N, Slope, LF, MSF$.

Table 1: Design Variable Distributions

Variable	Type	μ	σ	Nominal
ω_n	Normal	1975	25	1950
ζ	Log Norm	0.0025	0.0002	0.0025
ρ_{allow}	Normal	15	1.5	15
N	Normal	9200	100	9200
$Slope$	Normal	0.1	0.01	
LF	Normal	1	0.13	
MSF	Normal	1	0.13	
m	None	13		
Y	None	20%		

The distributions of the design variables are shown in Table 1. The "Type" column refers to the distribution type with mean μ and standard deviation σ. The nominal value is shown for those variables where it is needed and is usually the mean value. However the nominal blade natural frequency is not the mean due to the manufactured blades having a frequency biased from the design frequency. When the system variables are described as this, the problem has a probability of failure of 17.06%, as determined by a Monte Carlo analysis with one million realizations.

3 Stratified Sampling

Stratified sampling methods are a way to get a smaller confidence interval with a similar number of samples. If these methods can be optimized an even smaller confidence interval is produced for the same number of computations. The proposed method, Optimized Stratified Sampling (OSS), outlines an algorithm to accomplish this.

Probabilistic analysis predicts the probability of failure, p_f, hopefully including an estimate of the error, through integration of the joint probability distribution, $f(\underline{x})$, across the failure region, Ω:

$$p_f = \int_{\Omega} f(\underline{x}) d\underline{x} \tag{8}$$

Although there are several classes of methods to determine this probability, such as Fast Probability Integration FPI methods [8] or Mean-Based Reliability Methods [10], Fox shows that there is not an established method of estimating the error for these classes [3]. This is a major detriment since an analysis designed to a small margin may be an unsafe design if there is no error estimation. Sampling methods are better since they are an unbiased failure probability estimator, converge to the answer, and provide error estimate as well.

3.1 Stratified Sampling Methods

Monte Carlo techniques require a large number of samples in order to converge to the correct probability. This precludes any analysis that has a long limit state function evaluation time. Stratified sampling methods converge much more quickly [9] by subdividing each design variable into n equal probability bins. Further the sample set is constrained to one sample per bin enforcing the exact distribution over each design variable. Samples appear proportional to the distribution of a converged Monte Carlo analysis. Therefore, these methods are also known as Quasi-Monte Carlo (QMC) methods [13]. QMC methods maintain the benefits of sampling methods (e.g. unbiased estimate) while converging more quickly.

An equivalent integration of Equation (8) is to integrate over the entire space, but multiplying by a step function, H, of the limit state equation, g, that defines failure. If the system fails the step function evaluates to one, otherwise the step function is zero. This transformation is in Equation (9).

$$\int f(\underline{x}) H\left(g(\underline{x})\right) d\underline{x} \tag{9}$$

Sample methods evaluate this integral by weighting individual sample points by their respective area representation, A, as shown Equation (10).

$$\sum_{i=1}^{n} H\left(g(\underline{x}_i)\right) A\left(\underline{x}_i\right) = \frac{n_f}{n} \tag{10}$$

Since the sample point locations are not known a priori, each point is assumed to have equal area representation. The entire probability area is equal to 1 with n sample points, so each weighting is assumed to be $\frac{1}{n}$. Since the weighting is no longer a function of the sample point it can be pulled out of the summation. The sum of the step function also collapses to the number of failure points, n_f, because all failures evaluate to one and non-failures are equal to zero. The overall estimate of the failure probability is therefore Equation (10).

All sampling methods use this same method to calculate the estimate of the failure probability regardless of the method to obtain the sample point locations. There are several methods to obtain those sample points such as Latin Hypercube Sampling, Distributed Hypercube Sampling, and the proposed Optimized Hypercube Sampling.

3.1.1 Latin Hypercube Sampling

Latin Hypercube Sampling (LHS), first outlined by McKay et al [1], uses only the constraint on the single sample point per bin. The failure probability estimate converges quickly with number of samples and is widely used for probabilistic analysis. The theory is similar to Gaussian integration in that the space is completely evaluated by weighted results from single points. The weight for each point is equal since the volume represented by each sample is randomly distributed. When the points are more equally distributed, the variation in hypervolume representation decreases and the equal weighting assumption becomes more valid.

3.1.2 Distributed Hypercube Sampling

Distributed Hypercube Sampling (DHS) [2] adds a second constraint, advancing the idea that the points must be well distributed by constraining the sample set when projected onto a two-dimensional face of the hypercube. The term well distributed was defined as having a low coefficient of variation, COV, of the minimum distance between sample points. The added constraint increases the quality of the response as can be seen in a decrease in the confidence interval size. However the distribution in the volume of the design

space is still not addressed, leaving all optimization to the surface projection. Additionally the COV is reduced, but all points may be closely grouped together and the equal area representation assumed by all sampling methods is not valid possibly leading to poor estimates.

3.1.3 Optimal Stratified Sampling

Making the set evenly distributed on the edges and surfaces of the hypercube shrinks the uncertainty as shown by DHS. It follows that if the same constraint is applied to the hypercube volume rather than edges the benefit would increase. Just as LHS ignores hypercube surface distributions, DHS ignores hypercube volume distributions, leaving the gap that Optimized Stratified Sampling (OSS) fills.

Additionally, the proposed method requires a constraint that the minimum distance between points should be equal to an optimum distance based on perfect equal hypervolume representation by each sample point. The optimum is determined by Equation (11), where n is the number of bins and m is the number of dimensions.

$$\begin{aligned} volume &= n^m \\ \frac{volume}{point} &= \frac{n^m}{n} \\ d_{opt} &= \frac{n}{\sqrt[m]{n}} \end{aligned} \tag{11}$$

The jump from volume per point to the distance is a mathematically imprecise leap, taking the m root of the volume per point to find the edge length a hypercube of that volume. However imprecise, the resulting set is well distributed and spans the design space.

One method of measuring the distribution fitness established by [2] is the covariance of the minimum distance between points. The fitness of sample sets generated by the proposed method is also evaluated by this method.

4 Evolutionary Programming Algorithm

The proposed method uses Evolutionary Programming to optimize the sample set since alternative methods fall short in at least one area.

The primary alternative optimization routine is a gradient based scheme. Although the landscape is smooth for small changes, larger shifts are highly nonlinear and the gradient method fails. Additionally, the computation time for the gradients is high and since gradient is only valid for small changes, many gradients must be calculated in the course of optimization. EP uses the smooth behavior to make small changes without the expense of gradient calculations.

Another option is to use complete deterministic search. No deterministic method could search the whole space, which for our sample problem of 100 bins and 7 dimensions has a huge number of combination to search. Specifically the number of possible solutions is n^m choose n, which grows quickly with respect to increases in either bins or dimensions. The number of evaluations for the EP method is dependent only on the population size and generation count. These can be held to acceptable levels to prevent excessive computation costs.

A third option is a partial deterministic search. However this does not use the information from previous evaluations to affect the following samples. Without that intelligence the method is more inefficient and would presumably produce less optimal results in general. EP uses the best previous set to build the next set from, not losing the benefit of previous calculations.

A piecewise partial deterministic search would find consecutive points by a partial deterministic search. This has a large drawback of removing degrees of freedom from the design until there is only a single location left for the last sample point regardless of that points fitness. EP allows movement in all points at each iteration, avoiding that problem.

The decision to use EP over other EA methods is based on the fact that they do not fail where these alternatives do. Although the simplicity can cause EP to become trapped in a local minimum, the search landscape is relatively smooth and monotonically increasing in fitness so that hindrance should not be in issue in this problem.

The basis of the EP method is that the child generation is generated using only mutation, leaving out recombination and repair operators. Additionally, it would seem that recombination would be detrimental to the sample set. The fitness function is dependent equally on all points in the set. Recombination would destroy these sets while mutation only effects a small percentage of the data.

4.1 Data Structure

Each chromosome is a two-dimensional array, $a_{n,m}$, with length n, the number of bins, and width m, the dimensionality of the design space. Each sample point

is a row with the integer bin number for that dimension in the corresponding column. In the example shown in Table 2, the first sample point would be found in bin 41 for the first dimension, bin 34 for the second, etc.

Table 2: Data Structure

	1	2	...	m
1	41	34	...	86
2	62	12	...	70
3	89	42	...	31
⋮	05	59	...	97
n	22	83	...	15

4.2 Operators

The algorithm structure is shown in Figure (2) as a flowchart beginning with initialization of the population by random creation. These members are evaluated by the objective function then each member produces several children by means of mutation. Note that no encoding or decoding functions are needed for fitness function evaluation in the algorithm since the genotype and phenotype are integer values of the same form. The children are evaluated with the best surviving to reproduce again. That loop continues until some generation count is reached and a final optimized solution exists. Each of these parts will be introduced and described in the following sections.

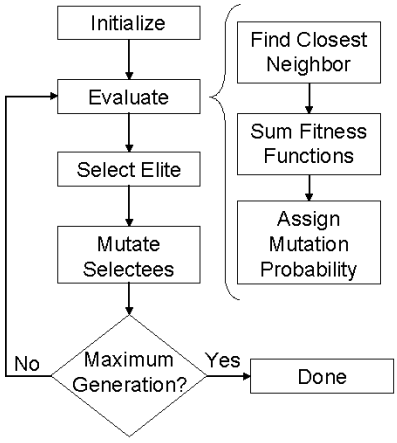

Figure 2: Operator Flowchart

The Evolutionary Programming parameter values used in the analysis can be found in Table 3. The small population size combined with elitism is an effort to best utilize the smooth response by continuing any better children. The fitness values are believed to converge after 500 generations, which is validated in the convergence subsection (5.2). The mutation rate is set to an expected single mutation per dimension. These parameters are applied to the one degree of freedom oscillator example so there are 10 bins and 7 dimensions.

Table 3: Evolutionary Programming Parameter Values

Parameter	Symbol	Value
Population size	μ	2
Children	λ	10
Generations	t_{max}	500
Mutation rate	p_m	1/n
Bin count	n	10
Dimensionality	m	7

4.2.1 Initialization operator

The initialization operator generates sample populations by randomly choosing the elements of each dimension of the array without repetition until that column in the data array is filled. The result is a feasible set of sample points that undergoes fitness evaluation and becomes the parents for all future members. Since the next steps in the algorithm are evaluation and selection, the number of initial population members is actually λ rather than μ. This allows for greater diversity in the initial population, hopefully meaning better initial fitness values.

4.2.2 Evaluation operator

The evaluation operation assesses the sample set using the objective function, Equation (12), which both minimizes the variance of the minimum distance between points. This is accomplished by minimizing the square difference between the optimal distance, Equation (11), and the minimum distance. Therefore a single objective function works to fulfill both objectives since making all the distances optimal also reduces the covariance of the distances as well.

$$\Phi(a(t)) = \sum_{i=1}^{\mu} \left(\min |a(t)_{i,:} - a(t)_{j,:}| - \frac{n}{\sqrt[m]{n}} \right)^2 \quad (12)$$

$$j = 1, \ldots, \mu \neq i$$

There are three main elements of the evaluation operation. The first aspect is finding the minimum distance to another sample location. The second takes utilizes that distance information and finds the fitness value for each population member. The third step is assigning a mutation probability to each point.

The first step, finding the closest point, is the single most computationally expensive aspect of the algorithm, so efforts are taken to reduce this cost. For example, when the distance from one point to another is calculated, that information is stored for use calculating the reverse distance, roughly reducing the computation time by half.

The second step is evaluating the fitness value, a function of the distances found in the first step. The objective function, Equation (12), is the sum of the square of the difference of the minimum distance and the optimum distance.

The third step is using the minimum distance information to define a probability of mutation to each sample location. The goal is to have a more fit function, so points are assigned a mutation probability based on the amount it adds to the fitness function. The probability comes from quadratic dynamic scaling. Each point is fit to a function such that the least fit function has a unit value and the most fit has zero probability. Further, the slope at the most fit point is zero so all points with values close to the most fit have near zero probabilities as well.

4.2.3 Mutation Operator

The mutation operation, a simple transposition operation defined by Simões in [7], varies the parent population creating the children though asexual reproduction. Each surviving parent produces the same number of children implying λ, number of children, must be a multiple of μ, number of parents. The overall probability of mutation is the inverse of the cell number, making the expected value of the binomial distribution to be one mutation per dimensions. Once the number of discrete swapping events is defined by selection from a binomial distribution, the points at which the swapping occurs is chosen probabilistically based on the quadratic scaling done in the evaluation step and the values are exchanged.

4.2.4 Selection Operator

The selection operation is of $(\mu + \lambda)$ formation, grouping the parent and child generations together for selection. They are ranked based on the fitness function and the μ fittest members are deterministically selected for survival. Other selection methods such as weighted random selection, but forcing the solution through elitism seemed to have few detriments for the faster convergence.

5 Results and Analysis

The heart of the matter is whether the proposed method produces better results than previous methods in an acceptable runtime [1]. Convergence is a very important aspect of the algorithm, for if the EA did not converge or converged to an unacceptable fitness value the results gained from that sample set are of limited value. The benefit is shown by hypothesis testing and analysis of variance.

5.1 Experimental Design

The experiment is designed to show the benefit of the new technique over other alternatives as it would be applied in an actual design problem. The procedure begins with the calculation of the sample points in the Optimized Stratified Sampling. Individual sets of n samples points define a sample design. Each design is evaluated several times to account for the random locations in the selected sample bins. The standard deviation of the probability estimates provides the information for the confidence interval. The experiment will compare the Coefficient of Variation of the minimum neighbor distances and the standard deviations of failure probability estimates for the optimized designs versus Latin Hypercube Sampling.

5.2 Convergence

Convergence, visualized by plotting the fitness versus generation, generally slows as the fitness improves since it takes longer to find a better solution. Figure (3) presents the fitness for the minimum maximum and mean convergence for the ten designs versus the logarithm of the generation. The logarithm is used because it allows changes in the convergence rate to be seen more easily.

Figure (3) shows that towards the end of the analysis the slopes generally decrease showing that the solutions have converged. Convergence is not sufficient for a good solution because an algorithm can converge to a suboptimal solution. For example, if the mutation rate is too high, the better sample sets might not be found because too many mutations occur.

To check if the converged solution has reached a value near that which was hoped, the COV values are compared to the COV from DHS. They range from 0.003 to 0.016 as compared with 0.083 for DHS [2], so the convergence value seems appropriate. The minimum,

[1] The algorithm was run on a Hewlett-Packard Pavilion desktop computer running Windows98 with a 550MHz Pentium II processor with MMX and 64MB RAM.

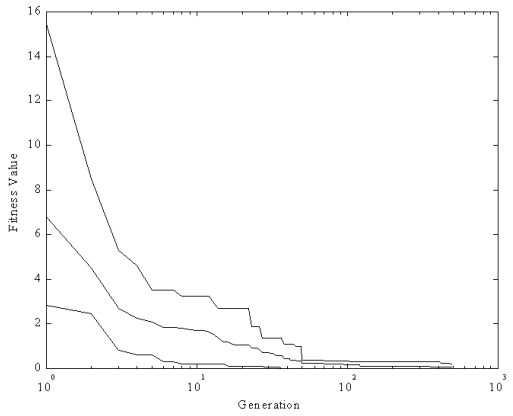

Figure 3: Fitness Convergence

maximum, and mean value plot of the COV is shown in Figure (4).

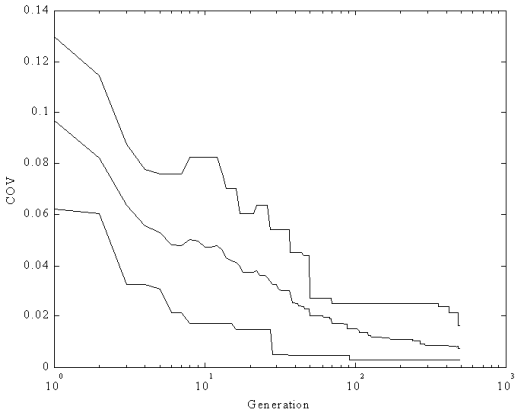

Figure 4: COV Convergence

5.3 Confidence Interval

The example problem used two methods for prediction of failure probability: Latin Hypercube and Optimized Stratified Sampling. DHS was not used because the algorithm was not available to the author, although results are compared to metrics published previously. All methods use 1000 function evaluations, 100 runs (10 designs, 10 times each) of 10 evaluations each. The goal is to determine if the new method reduces the size of the confidence interval.

Equation (13) generates a confidence interval based on repeated estimates of the mean as can be found in any standard basic statistics book [12]. The standard deviation of estimates of the mean is σ and n is the number of estimates.

$$\text{Conf} = 1.96\frac{\sigma}{\sqrt{n}} \tag{13}$$

Table 4 shows the final estimate of the mean and the associated confidence interval for the different methods after 1000 limit state function evaluations.

Table 4: Result Data

Statistic	LHS	OSS
Mean (μ)	17.5%	18.0%
Conf. Int.	2.60%	2.27%

The change from LHS to OSS lead to a 13% decrease in Confidence Interval. To determine if this decrease is meaningful or likely due to random variations an Analysis of Variance must be applied.

5.4 ANOVA Results

The Analysis of Variance (ANOVA) shows that there is a significant difference in the OSS over the LHS method. The ratio of the variances for the two methods is compared to the F tables to see if the difference is significant. Both the method variances have ninety-nine degrees of freedom based on the one-hundred statistics minus one for the mean probability of failure. The F-value, Table 5, corresponds to a confidence level for the null hypothesis that the variances are different of 91.1% confidence.

Table 5: ANOVA Values

Source	σ^2	F-Value
OSS	1.345E-2	1.312
LHS	1.765E-2	

The significant reduction in variance is due to the good distribution of sampling points in the design space through use of EP. The better distribution improves the assumption of equal weighting of sample points. The COV of the OSS method is significantly lower than published values for the DHS, even with the optimization applied to the volume rather than the surface of the design space.

6 Conclusions

The proposal of Optimized Stratified Sampling for choice of sample points follows the current trend in stratified sampling methods of imposing more constraints on the sample set so the points are more

evenly distributed. The goal of even distribution is to make the equal weighting assumption of the integration method more valid. Previously there was no method of optimizing the sample set within a design space that is too big to search deterministically and has no gradient information. Evolutionary Programming acts as an enabling technology that allows optimization where it could not be before.

The method implemented on the standard probabilistic analysis problem of a one-degree of freedom model of harmonic resonance converged to a distribution that is much better than alternate methods. This results in a better estimate of the probability of failure and s smaller confidence interval about that estimate.

It seems that the landscape assumption is not invalidated by the data. Elitism helps convergence and does not seem to produce traps at local minima.

Many times the cost of performing function evaluations in a probabilistic analysis is extremely high, almost prohibitive. The method described in this paper makes an effort to get the most information out of the evaluations that are performed. This method uses a small amount of computation before the probabilistic evaluation to achieve an improved confidence interval with acceptable computation times.

The algorithm is written in MATLAB and runs very quickly on moderate to high-speed desktop computers making it only a minimal cost to the developer. The recommendation for future analysis would be to run the algorithm on a fast desktop system. Current processors often outperform many SGI systems for single processor calculations. A moderate speed (1GHz) Pentium III or better processor often outperforms 300MHz SGI R12000 processors for single processor computation.

There is still an opportunity for improvement on the algorithm. The proposed method minimizes the variance of minimum distances in the volume of the cube and enforces uniform sampling in each dimension. It should be possible to enforce uniform distribution on each surface of the hypercube as well. This would attempt to insure that the projections onto each lower dimension are distributed as well as possible instead of just the highest and lower level.

References

[1] M. D. McKay, R. J. Beckman, & W. J. Conover, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," in *Technometrics*, 21(2), pp. 239-245, 1979.

[2] R. D. Manteufel, "Distributed Hypercube Sampling Algorithm," in *Proc. of 42nd AIAA Structures, Structural Dynamics, and Materials Conference*, 16-19 April, 2001.

[3] E. P. Fox, "On the Accuracy of various Probabilistic Methods," in *Proc. of 41st AIAA Structures, Structural Dynamics, and Materials Conference*, 3-6 April, 2001.

[4] J. Griffiths, C. Annis, Blair, B. Morris, K. Cornet, D. Ghiocel, and Nelson, "Probabilistic HCF Blade Design Progress Report," in *USAF Contract F33615-98-C-2928*, February, 2001.

[5] S. Rao, *Mechanical Vibrations, 2nd Ed.*, Reading, Massachusetts: Addison-Wesley, 1990.

[6] T. Bäck, *Evolutionary Algorithms in Theory and Practice*, New York, NY: Oxford University Press, 1996.

[7] A. Simões, "Using Genetic Algorithms with Sexual or Asexual Transposition: a Comparative Study," in *Proceedings of the Congress on Evolutionary Computation (CEC 2000)*, July 2000.

[8] B. H. Thacker, D. S. Riha, H. R. Millwater, M. P. Enright, "Errors and Uncertainties in Probabilistic Engineering Analysis," in *Proc. of 42nd AIAA Structures, Structural Dynamics, and Materials Conference*, 16-19 April, 2001.

[9] M. Stein, "Large Sample Properties of Simulations Using Latin Hypercube Sampling," in *Technometrics* Vol. 29(2), pp. 143-151, May 1987.

[10] B. M. Kwak & T. W. Lee, "Sensitivity Analysis for Reliability-Based Optimization using an AFOSM Method," in *Computers and Structures* Vol. 27, No. 3, pp.399-406,1987

[11] M. L. Shooman, *Probabilistic Reliability : An Engineering Approach*, Melbourne, FL: Krieger Publishing Company, 1990.

[12] P. G. Hoel, S. C. Port, & C. J. Stone, *Introduction to Statistical Thoery*, Boston, MA: Houghton Mifflin Company.

[13] D. Robinson & C. Atcitty, "Comparison of Quasi- and Pseudo-Monte Carlo Sampling for Reliability and Uncertainty Analysis," *AIAA–99–1589.*

[14] C. Annis, "Bayesian Network Analysis of the 1-D Oscillator Problem," *Independent Contractor Agreement 01-S441-48-01-C4*, http://www.StatisticalEngineering.com, 2001.