2015

A Distributed Surrogate Methodology for Inverse Most Probable Point Searches in Reliability Based Design Optimization

James Davidson
Wright State University

A Distributed Surrogate Methodology for Inverse Most Probable Point Searches in Reliability Based Design Optimization

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering

by

James M. Davidson
B.S.M.E., Wright State University, 2012

2015
Wright State University

Wright State University
SCHOOL OF GRADUATE STUDIES

April 30, 2015

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY James M. Davidson ENTITLED A Distributed Surrogate Methodology for Inverse Most Probable Point Searches in Reliability Based Design Optimization BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Engineering.

Ha-Rok Bae, Ph.D
Thesis Director

George Huang, Ph.D., P.E.
Chair, Department of Mechanical and
Materials Engineering

Committee on
Final Examination

Ha-Rok Bae, Ph.D

Ahsan Mian, Ph.D

Zifeng Yang, Ph.D

Robert E.W. Fyffe, Ph.D.
Vice President for Reseach and
Dean of the Graduate School

ABSTRACT

Davidson, James. M.S.Egr., Department of Mechanical and Materials Engineering, Wright State University, 2015. *A Distributed Surrogate Methodology for Inverse Most Probable Point Searches in Reliability Based Design Optimization.*

Surrogate models are commonly used in place of prohibitively expensive computational models to drive iterative procedures necessary for engineering design and analysis such as global optimization. Additionally, surrogate modeling has been applied to reliability based design optimization which constrains designs to those which provide a satisfactory reliability against failure considering system parameter uncertainties. Through surrogate modeling the analysis time is significantly reduced when the total number of evaluated samples upon which the final model is built is less than the number which would have otherwise been required using the expensive model directly with the analysis algorithm. Too few samples will provide an inaccurate approximation while too many will add redundant information to an already sufficiently accurate region. With the prediction error having an impact on the overall uncertainty present in the optimal solution, care must be taken to only evaluate samples which decrease solution uncertainty rather than prediction uncertainty over the entire design domain. This work proposes a numerical approach to the surrogate based optimization and reliability assessment problem using solution confidence as the primary algorithm termination criterion. The surrogate uncertainty information provided is used to construct multiple distributed surrogates which represent individual realizations of a lager surrogate population designated by the initial approximation. When globally optimized upon, these distributed surrogates yield a solution distribution quantifying the confidence one can have in the optimal solution based on current surrogate uncertainty. Furthermore, the solution distribution provides insight for the placement of supplemental sample evaluations when solution confidence is insufficient. Numerical case studies are presented for comparison of the proposed methodology with existing methods for surrogate based optimization, such as expected improvement from the Efficient Global Optimization algorithm.

List of Symbols

$g(\cdot)$	—	Limit State Response
$\mathbb{P}_f$	—	Probability of Failure
β	—	Reliability Index
x	—	Point in Physical Space
u	—	Point in Standard Normal Space
$x*$	—	Most Probable Point of Failure or Optimal Point (X-space)
$u*$	—	Most Probable Point of Failure (U-space)
u^*_{inv}	—	Inverse Most Probable Point of Failure (U-space)
x^*_{inv}	—	Inverse Most Probable Point of Failure (X-space)
n_{pc}	—	Number of Probabilistic Constraints
d	—	Deterministic Design Variables Set
X	—	Random Design Variables Set
P	—	Random Parameters Set
s	—	Probabilistic Constraint Shifting Vector
ρ_p	—	Distance Between Two Spacial Points
D	—	Vector of Sample Points
Y	—	Vector of Sample Responses
n_{dof}	—	Number of Degrees of Freedom
n_{ts}	—	Number of Truth Model Samples
d_w	—	Weighted Distance Between Two Spacial Points
$\mathcal{E}_d$	—	Approximation Stochastic Error
R_s	—	Observed Data Correlation Matrix
R_p	—	Prediction Data Correlation Matrix
μ	—	Gaussian Process Baseline Mean
σ^2	—	Gaussian Process Variance

$\hat{y}$	—	Surrogate Response
$\mathcal{E}_{cv}$	—	Cross Validation Prediction Error
$\mathcal{E}_{pe}$	—	Percent Residual Prediction Error
$\mathcal{E}_{sr}$	—	Standardized Residual Prediction Error
s^2	—	Standard Error of Gaussian Process Response
$E\left[I\right]$	—	Expected Improvement Response
L_{int}	—	Local Region Interval
L_{lb}	—	Local Region Lower Bound
L_{ub}	—	Local Region Upper Bound
V_v	—	Highest Error Vicinity Vectors Set
γ	—	Angle from Highest Error to Vicinity Point
μ_{x^*}	—	Mean Location of Distributed Solutions
σ_{x^*}	—	Location Standard Deviation of Distributed Solutions
$\mu_{\hat{y}^*}$	—	Mean Response of Distributed Solutions
$\sigma_{\hat{y}^*}$	—	Response Standard Deviation of Distributed Solutions

Contents

List of Figures

List of Tables

Acknowledgment

I would like to thank those who made this endeavor possible. Firstly, this includes my advisor, Dr. Ha-Rok Bae. His mentoring and guidance during my graduate studies has been invaluable. I would also like to thank the research group of which I have been extremely lucky to be a part, the Center of Excellence for Product Reliability and Optimization (CEPRO). Much of the knowledge I have gained has been a result of discussions with my colleagues in CEPRO and from its director, Dr. Ramana Grandhi.

This research was supported by Caterpillar, Inc. and Wright State University Research Initiation Grant. This support is gratefully acknowledged. In addition to funding this research, the design team at the Caterpillar Champaign Simulation Center was a valuable resource in determining how best to perform many of the software development aspects of this work.

Dedicated to my parents.

Introduction

In modern engineering design, computer simulations offer a cost effective and controlled method for the computation of system responses. This aides in the discovery of how a structural component may fail under certain operating conditions and which system parameters have the most influence for avoiding of such failures. Additional considerations must also be made regarding the uncertainties present in the system via loading, geometrical, and material properties. Traditionally, a safety factor would be applied to a component accounting for such uncertainties, however, determination of a probabilistic reliability against failure has proven to be a more useful method of dealing with system uncertainty. For such reliability assessment, distributions for each input parameter are propagated through the system, yielding a distribution for performance measures from which the probability of success can be computed.

In structural reliability based design optimization (RBDO), multiple methodologies and problem formulations exist for the determination of an optimum design which maintains a requested level of reliability. Many of these formulations require a most probable point (MPP) search using the first order reliability method (FORM) in a forward or inverse reliability assessment. Literature in the last decade has awarded more attention to the inverse reliability assessment variant, referred to as the performance measure approach (PMA) opposed to the forward variant, the reliability index approach (RIA). Initially, the reason for

this was due to the relative ease of locating the inverse MPP opposed to the forward MPP regarding the formulation of the optimization problem performing the search for each. More recently, the increased attention has been a consequence of decoupled RBDO methodology popularity for which a constraint shifting vector is normally computed using an inverse MPP.

Regardless of the methodology used for RBDO, the computational expense is often considerable since the computer simulations driving it typically consume large amounts of computer resources, taking hours or even days to complete the determination of a single response. Though using a decoupled RBDO method generally reduces the overall number of simulation responses necessary to be computed, for many problems the computational expense is still too overwhelming. To relive this expense, surrogate models are commonly used in place of simulations to drive iterative analysis procedures. In order to maximize the computational cost savings, very few responses may be sampled from the truth models for which the surrogates must be built upon. In such cases, the accuracy of the surrogate and solutions gained by its use must be verified. If the accuracy is found to be insufficient, supplemental evaluations may be required from the expensive model to improve the approximation. The location of these supplemental evaluations in the design space is critical due to truth model evaluation overhead and the risk of little improvement if a poor supplemental point is selected. This work proposes a method for the surrogate based determination of an inverse MPP for application in decoupled RBDO formulations. Due to the nature of the inverse MPP search, it is advantageous to first model this problem as global optimization.

The first part of the proposed method is concerned with the gathering of useful data which can be used to approximate the uncertainty associated with the most optimal solution the surrogate is capable of providing based on the prediction uncertainty. This involves the formation of multiple surrogates, referred to as distributed surrogates, all of which are based

on the prediction parameters of the initial approximation. Each distributed surrogate represents one particular behavior which the true response could potentially exhibit. After formation, each distributed surrogate is optimized upon to yield a set of distributed solutions. The spread in these solutions is the quantity used to approximate the uncertainty of the current solution at the centroid of the distributed solutions. A large spread in the distributed solutions is evidence supporting the need for more surrogate building information to obtain an optimal solution high in confidence. This spread in the distributed solutions is used as the primary termination criteria for the global optimization process.

The second part of the proposed method is concerned with the selection of one or more supplemental point locations, also referred to as infills, in the event that solution confidence is not sufficient. Since computational expense is invested in determining the distributed solutions, it is beneficial to include this information as criteria in the supplemental sampling procedure. Additional criteria is gained through the gathering of supplemental candidate points. Each candidate is determined through the local maximum search over the prediction uncertainty. A distance metric between each distributed solution region and each candidate is measured and used for the final determination of the supplemental point locations. Using such an infill criteria ensures that for spread out distributed solutions, a greater number of infills will be selected, reducing the overall number of algorithm iterations required through rapidly decreasing optimal solution uncertainty.

The remainder of this thesis is organized as follows. Background theory is then presented for forward and inverse reliability assessment along with the RBDO formulation using each, and surrogate based optimization. A literature review is then discussed concerning past contributions which address surrogate based optimization and inverse reliability assessment. The distributed surrogate optimization methodology is then described in detail with numerical examples following for both global optimization and inverse reliability assessment.

Background Theory

2.1 Reliability Assessment

The reliability of a system is the probability of successful performance under uncertain conditions. If the uncertain conditions impacting system responses can be characterized by a continuous frequency of outcomes, then the reliability of a system may be computed using a probabilistic approach. Utilizing probabilistic design methods can be used to reduce risk of failure in structural components due to uncontrollable variation in factors such as loading, geometry, material properties, manufacturing processes, and operation environments. The counterpart, deterministic design neglects system uncertainties, instead relying on safety factors to account for uncertainties. This section describes the traditional methods used to assess structural reliability.

For some uncertain variable outcomes, a system may not be able to perform as required. The performance requirements are expressed using a limit state function for which responses exceeding a specified value indicate failure. Using this formulation the margin of safety, or lack thereof, is quantified between the resistance and loading of a structure. The limit state, $g(\cdot)$, is therefore defined as

$$g(x) = R(x) - S(x) \tag{2.1}$$

where R is system resistance, S is system loading, and x is a vector of random variable outcomes. A limit state response can only be computed using an individual realization from the probability density of each random variable representing system uncertainties. Given this, the probability of failure, $\mathbb{P}_f$, corresponds to the likeliness of a less than zero limit state response expressed as

$$\mathbb{P}_f = \mathbb{P}\left[g\left(X\right) < 0\right] = \int \dots \int f_X\left(X_1, \dots, X_n\right) dX_1 \dots dX_n \tag{2.2}$$

where X is a vector of random variables and $f_X\left(X_1, \dots, X_n\right)$ is the joint probability density function for all variables in X. This leads to three possible limit state outcomes for different combinations of random variable realizations. Variable domain regions are classified for each outcome combination as $g\left(x\right) < 0 \rightarrow$ failure region, $g\left(x\right) = 0 \rightarrow$ failure surface, and $g\left(x\right) > 0 \rightarrow$ safe region. The integration in 2.2 is performed over the failure region. In most cases, $f_X\left(\cdot\right)$ is very difficult to integrate or not obtainable altogether as discussed in Mahadevan (2000) [26]. Therefore, many methods have been proposed to approximate it. The following subsections are devoted to describing the most applicable of these methods.

2.1.1 Mean Value Method

The mean value method is a straight forward, very inexpensive way of approximating the probability of failure of a system. Often, this method is also called the First-Order Second Moment (FOSM) or Mean Value FOSM (MVFOSM) since it uses the first-order terms of the Taylor series expansion at the mean value of each input variable and requires up to the second moments of the uncertain variables. Expanding the limit state function about the mean values and retaining the terms up to the first order provides a linearization of the

failure surface expressed as

$$\tilde{g}(X) \approx g(\mu_X) + \nabla g(\mu_X)^T \left(X_i - \mu_{X_i}\right) \tag{2.3}$$

where μ_X is a vector of the random variable means,

$$\mu_X = \{\mu_{X_1}, \ldots, \mu_{X_n}\}^T \tag{2.4}$$

and $\nabla g(\mu_X)$ is the gradient of the limit state at the mean value,

$$\nabla g(\mu_X) = \left\{ \frac{\partial g(\mu_X)}{\partial X_1}, \ldots, \frac{\partial g(\mu_X)}{\partial X_n} \right\}^T. \tag{2.5}$$

These expressions are used to approximate the distribution parameters of the limit state as

$$\mu_{\tilde{g}} = g(\mu_X) \tag{2.6}$$

and

$$\sigma_{\tilde{g}} = \left[\sum_{i=1}^{n} \left(\frac{\partial g(\mu_X)}{\partial X_i} \sigma_{X_i} \right)^2 \right]^{\frac{1}{2}} \tag{2.7}$$

where $\mu_{\tilde{g}}$ is the approximate limit state response mean and $\sigma_{\tilde{g}}$ is the approximate limit state response standard deviation. The formulation in equation 2.7 assumes all input random variables are statistically independent. Using this method of propagating the uncertainty through the system, a reliability index, β, can be computed as

$$\beta = \frac{\mu_{\tilde{g}}}{\sigma_{\tilde{g}}} \tag{2.8}$$

which can further be used to compute the failure probability as

$$\mathbb{P}_f = \Phi(\beta) \tag{2.9}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function.

Since this method utilizes a linear approximation of the limit state, it will not provide accurate reliability estimates for highly nonlinear limit state surfaces. Although second order terms could be added to the taylor series expansion to improve the approximation, the increase in computational effort is generally not proportional to the increase in accuracy.

2.1.2 MPP-Based Methods

The reliability index computation can also be expressed as an optimization problem wherein a Most Probable Point (MPP) of failure is sought. In standard normal variable space, the MPP is the point occurring on the failure surface which is located nearest to the origin. The Hasofer-Lind (HL) algorithm, introduced by Hasofer and Lind (1974) [19], performs a linear transformation of the random design vector X into U, a vector of standardized independent normally distributed variables. This yields a symmetrical joint PDF centered at the origin in U-space. With this transformation, it becomes fairly intuitive that the point of largest probability density in U-space with a limit state response of zero has the most significant contribution to the probability of failure.

The transformation from a point in physical space to a point in standard normal space is achieved using the distribution parameters of each random variable. For random variables which are already normally distributed, the forward transform for an X-space point, x, to

U-space point, u, is expressed as

$$u = T\left(x, X\right) = \frac{x - \mu_X}{\sigma_X} \tag{2.10}$$

with the inverse transformation expressed as

$$x = T^{-1}\left(u, X\right) = \left(u\right)\left(\sigma_X\right) + \mu_X \ . \tag{2.11}$$

If the X-space variables are not normally distributed, an extra step must be used to determine the equivalent normal approximation such as the Rosenblatt transform, described by Rosenblatt (1952) [34], or Nataf transform, generalized in Lebrum (2009) [23]. Using the inverse transformation, the optimization formulation for the MPP search is expressed as

$$\begin{aligned} &find && u^* \\ &minimize && \|u\| \\ &subject\ to && g\left(T^{-1}\left(u, X\right)\right) \leq 0 \end{aligned} \tag{2.12}$$

where u^* is the MPP. The reliability index can now be expressed in terms of the MPP as

$$\beta = \|u^*\| \ . \tag{2.13}$$

This method of determining the reliability which corresponds to a response level in the limit state is referred to as the reliability index approach (RIA).

Another method, the performance measure approach (PMA), determines the location of an inverse MPP. Here, the probability of success is specified, and the corresponding response level found. The performance measure approach optimization formulation for the inverse

MPP search is expressed as

$$\begin{aligned} &find && u^*_{inv} \\ &minimize && g\left(T^{-1}\left(u, X\right)\right) \\ &subject\ to && \|u\| = \beta^t \end{aligned} \tag{2.14}$$

where u^*_{inv} is the inverse MPP and β^t is the specified target reliability. The goal of the operation shown in equation 2.14 is to minimize the limit state response on the surface of a hyper-sphere which has a radius equal to the reliability index.

For both RIA and PMA, different methods can be used to approximate the response of the limit state function for the optimization process if the response evaluations are particularly expensive. The usage of a first order approximation at the MPP is refereed to as first order reliability method (FORM), whereas if the second order terms are included in the approximation, it is called second order reliability method (SORM).

Although the constrained optimization problems in equations 2.12 and 2.14 can be solved using many general purpose numerical optimization algorithms including gradient based, feasible directions, and penalty methods, iterative search methods developed to solve the reliability problem may offer advantages for some problems. One of these methods is the HL algorithm and it's version later extended by Rackwitz and Fiessler (1978) [33] (HL-RF) to incorporate the steps required for non-Gaussian random variables. As in RIA, there are special algorithms to perform PMA and locate the inverse MPP such as Advanced Mean Value (AMV) and Hybrid Mean Value (HMV) described by Youn (2004) [39].

2.1.3 Sampling Methods

A more experimental approach for computing the failure probability is available using sample methods. These methods hold the advantage of being simple to implement and providing accurate probabilistic information for sufficiently large sample sizes since they are numerical integrating technique using direct response evaluations without approximations. Sampling methods propagate uncertainty through the system by evaluating the limit state response for many trial points whose degrees of freedom are selected according to their random distribution. Using an indicator function, the number of trials which fall within the safe region, not returning a negative limit state, are counted and divided by the total number of trials. This operation is expressed by

$$\mathbb{P}_f = \frac{N_f}{N} \tag{2.15}$$

where N_F is the number of trial points which resulted in a violated limit state response and N is the total number of trials. However, the number of trails which must be used for accurate results is generally very large. For response evaluations which are computationally expensive, the usage of sampling methods become unpractical. The usage of monte carlo simulations and Latin hypercube sampling are two popular methods for selecting the trial points.

Points sampled using a Monte carlo simulation consist of a set randomly drawn from each random variable distribution. When randomly drawing number from a probability density, it is obvious that the vast majority will reside somewhat near the mean, leaving the distribution tail regions sparsely populated. This can be problematic for reliability analysis since low probabilities are computed when few of the trial points evaluated have extreme values. To avoid error due to the nature of random sampling, very large sample sizes must me selected.

Stratified sampling methods such as Latin hypercube sampling maintain a certain degree of space filling properties. This improves the coverage of the random space for a smaller sample size and reduces the likeliness of an uneven sample set being generate. Latin hypercube sampling will be discussed in greater detail in the design of experiments section of this document when the selection of sampling points to build surrogates is discussed.

2.2 Reliability-Based Design Optimization

In traditional optimization, the design variables as well as all other information provided in the problem specification have no randomness, leading to a deterministic optimum. However, the attributes for a physical system will always have a degree of uncertainty. Deterministic design solutions will require compensation to reduce risk of failure during system operation, generally in the form of a safety factor or some other form of conservative optimum shift. If left unchanged, the optimal deterministic designs for symmetrically distributed uncertainties will generally have a reliability of nearly 50% since variations in system output will cause approximately half of the performance measures to violate their constraints.

In reliability based design optimization (RBDO), probabilistic constraints are applied to ensure a specified reliability against failure for each performance constraint. The reliability for each constraint must be assessed after design changes occur in an optimization loop. Over the past decade, many formulations have appeared in the literature for performing optimization and reliability assessment simultaneously. The following subsections are devoted to describing the formulations which are most applicable to the methodology which will be proposed later in this document.

2.2.1 Nested RBDO

The general problem formation for RBDO using a forward reliability assessment considering only probabilistic constraints can be defined as

$$\begin{aligned} &find && \{d, \mu_X\} \\ &minimize && f\left(d, X, P\right) \\ &subject\ to && \beta_j \geq \beta_j^t \qquad j = 1, 2, ..., n_{pc} \end{aligned} \tag{2.16}$$

where $f\left(\cdot\right)$ is the objective response, β_j is the reliability index for the i^{th} probabilistic constraint, β_j^t is the reliability index target for the j^{th} probabilistic constraint, d is a vector of the deterministic design variables, X is a vector of the random design variables, and P is a vector of the random parameters. Here, a vector of deterministic design variable values and mean values for the random design variables are sought, represented by $\{d, \mu_X\}$ and henceforth refereed to as the design set. The random parameters P have fixed mean values and therefore only introduce uncertainty into the system. The evaluation of each design set involves a reliability assessment for each probabilistic constraint. This is most easily implemented as a nested loop operation where RIA is used in the inner loop to determine the probability of success for each probabilistic constraint for the current design set given the associated system uncertainty. This inner process described by

$$\begin{aligned} &find && u_j^* \\ &minimize && \|u\| \\ &subject\ to && g_j\left(\{d, x_{rs}\}\right) \leq 0 \\ &for\ which && x_{rs} = T^{-1}\left(u, \{X, P\}\right) \end{aligned} \tag{2.17}$$

where $g_j(\cdot)$ is the limit state response for the j^{th} probabilistic constraint, u is a vector of ll random variables and parameters transformed to standard normal space, x_{rs} is the random set outcome in physical space corresponding to u, and u_j^* is the forward MPP for the j^{th} probabilistic constraint. The reliability index associated with the forward MPP defined as

$$\beta_j = \left\| u_j^* \right\| \tag{2.18}$$

for the j^{th} probabilistic constraint. The probability of limit state feasibility is equal to the probability associated with the resulting reliability index shown mathematically as

$$\Phi\left(\beta_j\right) = \mathbb{P}\left(g_j\left(\{d, x_{rs}^*\}\right) \geq 0\right) \tag{2.19}$$

for the current design set vector under evaluation in the outer optimization loop, where $x_{j,rs}^*$ is the physical space forward MPP for the j^{th} probabilistic constraint. Again, the convention used is that in which the feasible region for each probabilistic constraint is defined by a positive limit state response.

An alternative formulation for double loop RBDO using an inverse reliability assessment can be defined as

$$\begin{array}{ll} find & \{d, \mu_X\} \\ minimize & f(d, X, P) \\ subject\ to & g_j\left(d, x_{j,inv,rs}^*\right) \geq 0 \qquad j = 1, 2, ..., n_{pc} \\ for\ which & x_{j,inv,rs}^* = T^{-1}\left(u_{j,inv}^*, \{X, P\}\right) \end{array} \tag{2.20}$$

where $g_j\left(d, x_{j,inv,rs}^*\right)$ is the minimum limit state response for the current design under evaluation which corresponds to the target reliability index and $x_{j,inv,rs}^*$ is the physical

space inverse MPP corresponding to $u^*_{j,inv}$, the standard normal space MPP for the j^{th} probabilistic constraint. A PMA inverse reliability assessment is undergone within the inner loop after each design change for this formulation. This procedure is expressed as

$$\begin{aligned} &find && u^*_{j,inv} \\ &minimize && g_j\left(d,u\right) \\ &subject\ to && \|u\| = \beta^t_j \end{aligned} \tag{2.21}$$

for determination of the inverse MPP for each probabilistic constraint. RBDO with PMA is often considered more robust than RIA due to the relative simplicity of searching for the inverse MPP given the spherical location constraint in inverse reliability assessment versus searching for the MPP for a negative or zero limit state condition in forward reliability assessment. However, there are situations where RIA is more computationally efficient, such as when the probabilistic constraints are being violated as discussed in [36]. PMA holds the advantage when the constraints are not active.

2.2.2 Decoupled RBDO

Applying the double loop methods for RBDO may involve many iterations of the outer loop which must perform numerous reliability assessments for design sets which are far from feasible, driving the cost of such analysis unreasonably high. Considerable computational cost savings can be achieved by formulating the nested probabilistic optimization problem as a series of decoupled deterministic optimization and reliability assessment operations. In decoupled RBDO formulations such as the Sequential Optimization and Reliability Assessment (SORA) method intruduced by Du (2004) [13], a shifting vector is applied to the random set for the purpose of moving infeasible constraint boundaries into the feasible region. This shifting vector is selected by computing the inverse MPP for the most

recent design set from the deterministic optimization step with inverse reliability analysis. In SORA, the formulation for one cycle of deterministic optimization, equivalent to probabilistic optimization, is expressed as

$$\begin{aligned} &find && \{d^k, \mu_X^k\} \\ &minimize && f\left(d, X, P\right) \\ &subject\ to && g_j\left(d, \mu_X - s_j^{(k-1)}, x_{j,inv,P}^{*(k-1)}\right) \geq 0 \qquad j = 1, ..., n_{pc} \end{aligned} \tag{2.22}$$

where $s_i^{(k-1)}$ is the shifting vector computed in the previous cycle associated with the j^{th} constraint, $x_{j,inv,P}^{*(k-1)}$ is the physical space inverse MPP computed in the previous cycle with respect to only the random parameter portion of the random set associated with the j^{th} constraint, and k is the current SORA iteration. The shifting is vector computed using

$$s_j^{(k-1)} = \mu_X^{(k-1)} - x_{j,inv,X}^{*(k-1)} \tag{2.23}$$

where $\mu_X^{(k-1)}$ is a vector containing the mean values for the random design variables found by the optimizer in the previous cycle and $x_{j,inv,X}^{*(k-1)}$ is the physical space inverse MPP computed in the previous cycle with respect to only the random design variable portion of the random set. Here, one cycle of SORA contains the deterministic optimization to solve for the most optimum design set given the most recently computed shifting vector shown in equation 2.22 followed by inverse reliability analysis shown in equation 2.21 using the design set which was just determined to update the shifting vector for use in the next cycle. The mean values associated with the initial random set are used in place of the inverse MPP in the first iteration of SORA for both the deterministic optimization and computation of the shifting vector. The termination criteria for SORA is the convergence of the design set which is tested following each deterministic optimization operation during and after the second iteration.

The cost savings in decoupled RBDO is mostly due to the avoidance of many reliability assessment operations with only one of these being performed for each deterministic optimization. The cost savings in doing so are very significant when compared to double loop formulations as suggested by comparison studies presented in [2].

2.3 Surrogate-Based Optimization

The general process for surrogate based optimization involves the strategic selection of training points, building of a surrogate with quantification of prediction uncertainty, and the section of a supplemental sample in an area of interest in the design domain. This section provides an overview of popular methods to achieve these steps.

2.3.1 Design of Experiments

Surrogate based analysis requires a finite set of sample responses to be evaluated from the truth model for which to build the approximation model upon. The number of samples evaluated and their locations in design space necessary for adequate response approximation is problem dependent and generally unknown. The Design of Experiments (DOE) stage of surrogate based analysis is concerned with determining the best possible initial sampling plan given the number of degrees of freedom and the range of each degree which is under investigation. Without prior information about the system response, which is the assumption in the work presented here, the best sampling plan is often considered to be one which is optimally filling of the design space. This is important for two reasons, firstly, when using samples whose locations are equally distributed throughout the problem domain, it can be assumed the greatest possible amount of features from the truth model have been

captured, and secondly, it avoids local regions within the problem domain which have a high sampling density providing redundant information and possibly opening the door for surrogate model building problems down the road depending on the surrogate type selected.

Latin Hypercube Sampling (LHS) is a popular method for determining an initial sampling plan for surrogate modeling. It achieves domain stratification by dividing the range for each degree of freedom into sections with the number of intervals equal to the number of samples required. For the two dimensional case, where the intervals can be visualized as a grid of cells, a point would be chosen at random from each cell of a Latin Square which is determined by selecting exactly one cell in each row and column. A sample plan formed using this method guarantees the domain of each individual degree of freedom be fully represented, however the design space as a whole may not be optimally space filled. For instance, a group of cells formed by taking those which occupy the diagonal of our grid technically comprises a Latin square. Many extensions to LHS have been proposed to avoid poorly filling sample plans such as random orthogonal arrays [30], cascading Latin hypercubes [18], and optimal Latin hypercubes [16].

To determine the relative optimality with respect to space-fillingness for one sample plan versus another, criteria to measure this property must be derived. Firstly, the distance between any two points in space can be expressed by

$$\rho_p\left(x^{(i_1)}, x^{(i_2)}\right) = \left[\sum_{j=1}^{n_{dof}} \left| x_j^{(i_1)} - x_j^{(i_2)} \right|^p \right]^{\frac{1}{p}} \tag{2.24}$$

where ρ_p is the distance measure of the p^{th} order between two points in space represented by $x^{(i_1)}$ and $x^{(i_2)}$ and n_{dof} is the number of dimensions associated with each point. Here, $p \geq 1$ with $p = 1$ representing rectangular distance and $p = 2$ representing Euclidean distance. A sample plan which exhibits decent space-filling properties will contain points which are spread out, representing the entire design domain while having no two points

located too close together. An optimization formulation for determining a sample plan with this property to the highest degree can be formulated as

$$\begin{array}{ll} find & D_{sf} \\ maximize & \min_{x^i \in D} \left[\rho_p \left(x^{(i_1)}, x^{(i_2)} \right) \right] \end{array} \tag{2.25}$$

where D_{sf} is the optimal space filling design, referred to as the maxmin design, which maximizes the minimum distance between any two points in a trial Latin hypercube design, specified here as D. This criteria as well as other more sophisticated definitions required for thorough algorithm design are are discussed by Morris (1995) [29]. This search can be carried out in a number of ways, such as randomly creating a predetermined number of LHS designs and selecting the one which has the best properties according to equation 2.25. However, a superior approach to meet this end may be through the use of an evolutionary strategy as suggested by Forrester (2008) [16]. Applying an evolutionary strategy to this problem involves randomly mutating a promising sampling plan to varying degrees over a series generations while selecting the best resulting designs for further mutation. Algorithms of this type can have self adapting rates of mutation or linearly declining mutation rates in order to converge to a best design. The mutation operation for guaranteeing that each trial design maintains its Latin hypercube properties generally involves the random swapping of dimension values across different points in the sample plan.

The notation to describe the sampling plan with associated responses is as follows. A single point within the design space is described by

$$x = \left\{ x_1 \quad x_2 \quad \dots \quad x_{n_{dof}} \right\} \tag{2.26}$$

where n_{dof} is the dimensionality of the sampling space. The sampling plan resulting from

the optimal Latin hypercube procedure is described by

$$D_{sf} = \begin{Bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n_{ts})} \end{Bmatrix} \tag{2.27}$$

where n_{ts} is the number of samples specified to be selected. The truth model response vector at these locations are arranged as

$$Y = \begin{Bmatrix} y\left(x^{(1)}\right) \\ y\left(x^{(2)}\right) \\ \vdots \\ y\left(x^{(n_{ts})}\right) \end{Bmatrix} \tag{2.28}$$

where y is the truth response function.

2.3.2 Gaussian Process Modeling

Though many options are available for approximating a truth model using set of sampled responses, Kriging has become an especially popular surrogate modeling method which is a special case of a Gaussian process model. This method is selected by many due to its ability to both capture nonlinear truth model behavior and provide statistical information regarding the approximation. Another advantage is the fact that Kriging is an interpolation approximation with a zero mean squared error at training point locations.

In Kriging, the predicted response at a point is formulated as the sum of a mean regression baseline value and a stochastic "error" term which specifies the deviation of the approxi-

mated value from the baseline mean. These errors are assumed to be continuous and correlated with respect to the weighted distance between sample point locations. This weighted distance between two points is described by

$$d_w\left(x^{(i)}, x^{(j)}\right) = \sum_{h=0}^{n_{dof}} \theta_h \left| x_h^{(i)} - x_h^{(j)} \right|^{p_h} \qquad \left(\theta_h \geq 0, p_h \in [1,2]\right) \tag{2.29}$$

where θ_h and p_h can be thought of as the activeness and function smoothness, respectively, for the h^{th} design space dimension [21]. By using the weighted distances between all pairs of points, a matrix R_s containing the corresponding correlations for these point pairs based on the observed data can be formed. Each element of the matrix is defined by

$$\text{corr}\left[\mathcal{E}_d\left(x^{(i)}\right), \mathcal{E}_d\left(x^{(j)}\right)\right] = \exp\left[-d_w\left(x^{(i)}, x^{(j)}\right)\right] \tag{2.30}$$

so that

$$R_s = \begin{bmatrix} \text{corr}\left[\mathcal{E}_d\left(x^{(1)}\right), \mathcal{E}_d\left(x^{(1)}\right)\right] & \dots & \text{corr}\left[\mathcal{E}_d\left(x^{(1)}\right), \mathcal{E}_d\left(x^{(n)}\right)\right] \\ \vdots & \ddots & \vdots \\ \text{corr}\left[\mathcal{E}_d\left(x^{(n)}\right), \mathcal{E}_d\left(x^{(1)}\right)\right] & \dots & \text{corr}\left[\mathcal{E}_d\left(x^{(n)}\right), \mathcal{E}_d\left(x^{(n)}\right)\right] \end{bmatrix} \tag{2.31}$$

where $\mathcal{E}_d(x)$ is the approximation stochastic error at x.

The values θ and p for each design space dimension are selected to maximize the likelihood of the sample which is described by

$$L\left(\mu, \sigma^2\right) = \frac{1}{(2\pi)^{\frac{n}{2}} \left(2\sigma^2\right)^{\frac{n}{2}} |R_s|^{\frac{1}{2}}} \exp\left[-\frac{(Y - \mathbf{1}\mu)' R_s^{-1} (Y - \mathbf{1}\mu)}{2\sigma^2}\right] \tag{2.32}$$

where

$$\mu = \frac{\mathbf{1}' R_s^{-1} Y}{\mathbf{1}' R_s^{-1} \mathbf{1}} \tag{2.33}$$

and

$$\sigma^2 = \frac{(Y - \mathbf{1}\mu)' R_s^{-1} (Y - \mathbf{1}\mu)}{n} \tag{2.34}$$

with $\mathbf{1}$ being an n-vector of ones. Once all the optimal values for θ and p have been found, a predicted value at point x is formulated as

$$\hat{y}(x) = \mu + R_p' R_s^{-1} (Y - \mathbf{1}\mu) \tag{2.35}$$

where R_p is the vector containing the correlations between all sample and prediction point locations, described by

$$R_p = \begin{bmatrix} \text{corr}\left[\mathcal{E}_d(x), \mathcal{E}_d\left(x^{(1)}\right)\right] \\ \text{corr}\left[\mathcal{E}_d(x), \mathcal{E}_d\left(x^{(2)}\right)\right] \\ \vdots \\ \text{corr}\left[\mathcal{E}_d(x), \mathcal{E}_d\left(x^{(n)}\right)\right] \end{bmatrix}. \tag{2.36}$$

The overhead of building a Kriging model is normal subject to the dimensionality of the approximation and the method used to select the optimal hyper parameters θ and p. Many optimization algorithms have been used for the optimization of these parameters for maximization of equation 2.32. Since the maximum likelihood surface is generally multimodal with noncontinuous gradients, a global optimization technique is normally applied.

2.3.3 Surrogate Prediction Error

The quality of a surrogate model with respect to how well it represents the truth model, said to be the prediction error, can be calculated using multiple methods. Two popular measures of this error are the root mean squared error ($RMSE$) and the correlation coefficient (r^2). The formulation for computing the $RMSE$ is

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2}{n_{ts}}} \tag{2.37}$$

where y and $\hat{y}$ are the truth and surrogate model responses, respectively, for the i^{th} location in the design space among the n_{ts} samples. The formulation for r^2 is

$$r^2 = \left(\frac{\sigma\left(y, \hat{y}\right)}{\sigma_y \sigma_{\hat{y}}}\right)^2 \tag{2.38}$$

where $\sigma\left(y, \hat{y}\right)$ is the covariance between the truth and surrogate model responses, σ_y is the standard deviation of the truth model responses, and $\sigma_{\hat{Y}}$ is the standard deviation of the surrogate model responses. While it is desirable to obtain a model with the lowest $RMSE$ and r^2 as close to 1.0 as possible, in practice, a $RMSE$ value less than 10% of the range of model responses and a r^2 value of greater than 0.8 is normally associated with an approximation which has reasonable predictive capabilities [16]. To be used with interpolation approximations such as Kriging, it is necessary to withhold a certain number of sample points from the approximation training data to use for testing since the approximated responses at the training data points match the true responses exactly. Calculating the $RMSE$ and r^2 using responses from the surrogate and truth models which were used to train the surrogate will yield values of 0.0 and 1.0, respectively, which does not provide useful information regarding the accuracy of the interpolated regions between sampled points. Saving sample

points for calculating this type of prediction error is often considered a luxury which cannot be afforded when using data from an expensive computational model.

Another widely used method for calculating the prediction error of an approximation model is cross validation. In a cross validation procedure, the sampled points are randomly divided into q equally sized testing subsets [20]. The truth responses for each subset are then compared against the responses from a surrogate model generated from the combined data in the remaining $q-1$ subsets. The result is n response differences which can be associated with each corresponding sample point. The prediction error is expressed as the average of these differences by

$$\mathcal{E}_{cv} = \frac{1}{n} \sum_{i=0}^{n} \left[y^{(i)} - \hat{y}^{(i)} \left(X_t \right) \right] \tag{2.39}$$

with

$$X_t = \{ x \mid x \in X \wedge x \notin X_q \} \tag{2.40}$$

where X, X_q, and X_t are sets of sample points containing the total samples set, current cross validation subset which contains the i^{th} group of sample points, and current surrogate training set. The prediction error calculated by cross validation can be bias in regards to the selection of q. If $q = n$, the procedure is referred to as *leave one out* cross validation. Using this variant of cross validation can largely eliminate the bias related to specifying an alternative q value, but does increase the computational demands since n surrogates must be trained. However, the cost of creating the surrogate model is often considered negligible when compared to the expensive computational model.

Also obtainable during cross validation are two quantities which attempt to describe the amount of local variation in the model, and thereby the prediction error [20]. These are the percent residual error and standardized residual error. The percent residual error is the

more simple of the two to quantify and described by

$$\mathcal{E}_{pe} = \left| \frac{y^{(i)} - \hat{y}^{(i)}\left(X_t\right)}{y^{(i)} + \alpha} \right| \times 100 \tag{2.41}$$

where α is a small number. The standardized residual is described by

$$\mathcal{E}_{sr} = \frac{y^{(i)} - \hat{y}^{(i)}\left(X_t\right)}{s^{(i)}\left(X_t\right)} \tag{2.42}$$

where $s^{(i)}$ is the square root of the mean square error from Kriging at the point left out of the training set. The mean square error in terms of the Kriging variance and correlation matrices is

$$s^2\left(x\right) = \sigma^2\left[1 - R_p' R_s^{-1} R_p\right] \tag{2.43}$$

for a point x at any location within the approximation domain. Having these values at every sample point provides a means to compare the prediction variation and hopefully determine a location where the model could use improvement if the prediction error is found to be too large either globally or locally.

2.3.4 Expected Improvement Infilling

To globally optimize on a surrogate, an infilling criterion must be selected. Infill points, locations selected to be evaluation by the expensive truth model, are selected based on the potential gain of useful knowledge at points of particular interest and generally driven by two mechanisms - exploration and exploitation of the surrogate. Exploration of the surrogate is concerned with adding information to the surrogate building process for the improvement of prediction accuracy. Sequentially selecting infill points at the location of the maximum prediction variance would be an example of pure surrogate exploration.

Exploitation of the surrogate is concerned with searching for a location which provides the most optimal point. Sequentially selecting infill points at the location of the current best solution would be an example of pure surrogate exploitation.

A balanced goal seeking criterion to achieve both exploration and exploitation simultaneously is an attractive criterion attribute the for selecting infill point locations. One such method from the Efficient Global Optimization (EGO) algorithm proposed by Jones (1998) [21] is the Expected Improvement function. The expected improvement criterion is a function of both the uncertainty of a prediction and current best solution. This quantity is expressed as

$$E\left[I\left(x\right)\right] = \left(f_{min} - \hat{y}\right)\Phi\left(\frac{f_{min} - \hat{y}}{s}\right) + s\phi\left(\frac{f_{min} - \hat{y}}{s}\right) \tag{2.44}$$

for global minimization where f_{min} is the most optimal training sample response, $\hat{y}$ is the surrogate predicted response at x, s is the standard error of the predicted response, $\Phi\left(\cdot\right)$ is the standard normal cumulative distribution function, and $\phi\left(\cdot\right)$ is the standard normal probability distribution function. Equation 2.44 provides a measure of the how much the objective value at x is expected to be less than the current best solution predicted response.

For each iteration of EGO, multiple steps are taken as shown in figure 2.1. After initial sampling of the design domain and a a Kriging model trained, a global minimization upon the predicted response surface is be performed. The algorithm termination criterion is a predefined ratio between the current best surrogate response and the expected improvement value. As suggested by Jones (1998) [21], algorithm termination occurs when the maximum expected improvement value is greater than one percent of the current best solution. If this condition is not met, one additional training sample is added to the surrogate building set at the location of the maximum expected improvement.

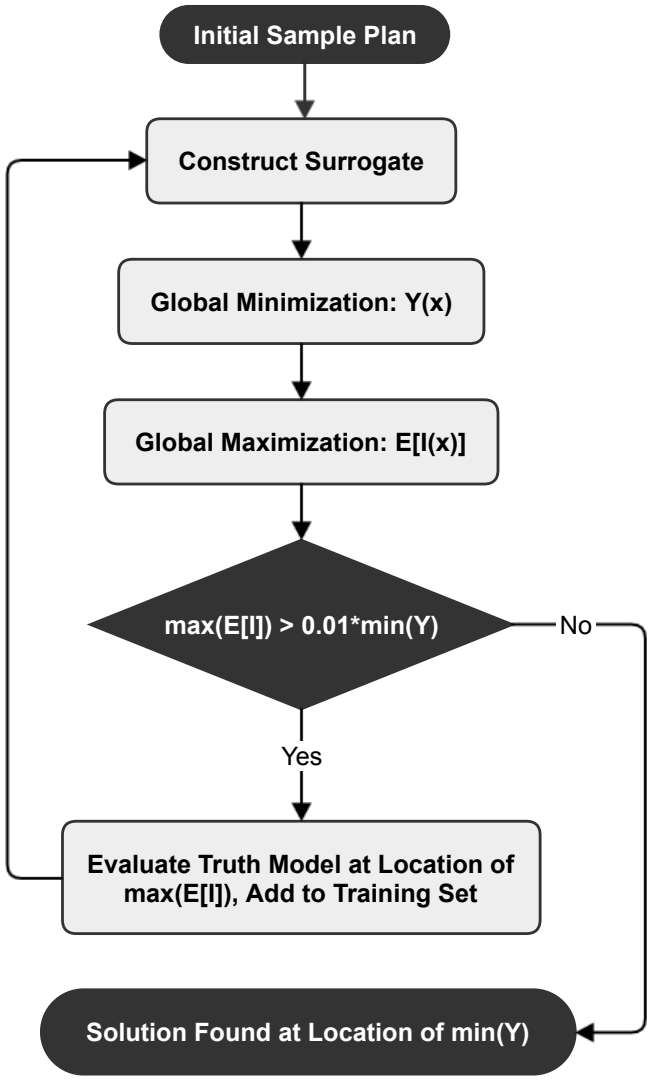


Figure 2.1: Expected Improvement Optimization Process Flowchart

A one degree of freedom optimization problem is solved using expected improvement to illustrate this process. The closed form truth model for this problem is expressed by

$$f(x) = \frac{x^2 + 10\sin(x)}{50} \in [-10, 10]. \tag{2.45}$$

This example, like all optimization demonstrations presented in this work, has been scaled between $[1, 0]$ for the range shown in equation 2.45. Figure 2.2 describes the true behavior

of the problem which will be minimized globally with indication of the true solution.

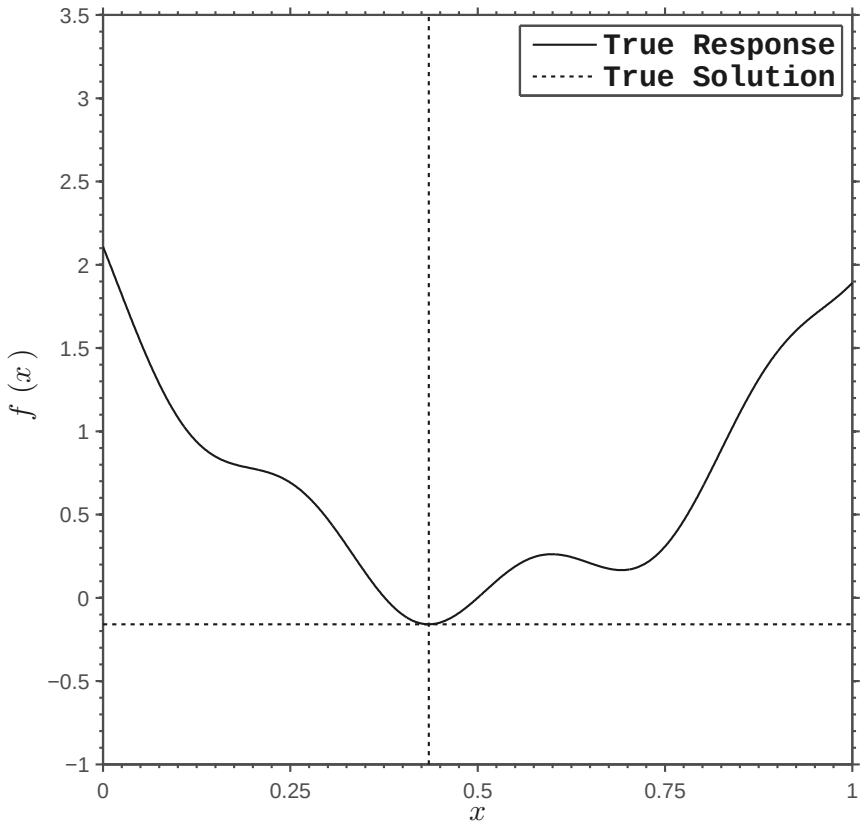
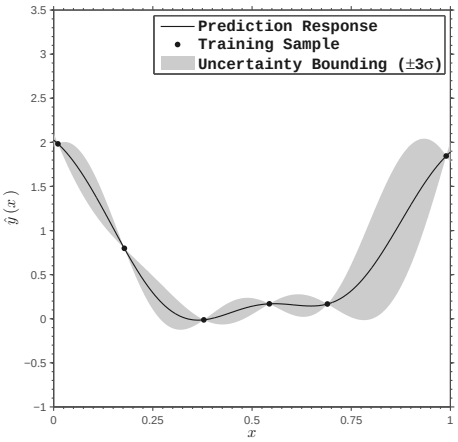
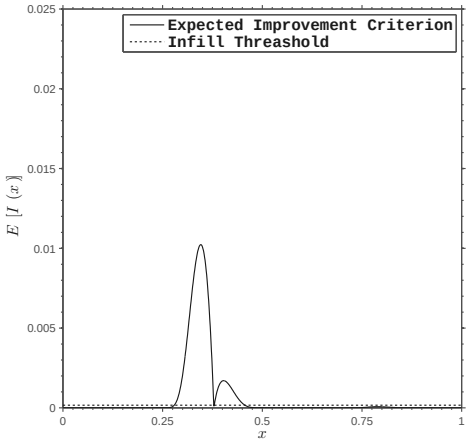


Figure 2.2: 1D Global Optimization Example Truth Behavior

Expected Improvement is shown to solve this problem using three infill samples for an initial Gaussian process trained using six truth samples. The algorithm progress is shown in figures 2.3 through 2.6. These progress figures indicate the largest expected improvement value was observed in iteration two, and after the selection of two additional infill samples, fell below the threshold value. The final solution, shown in figure 2.6a, determined by expected improvement for this problem agrees with the true solution with negligible error.

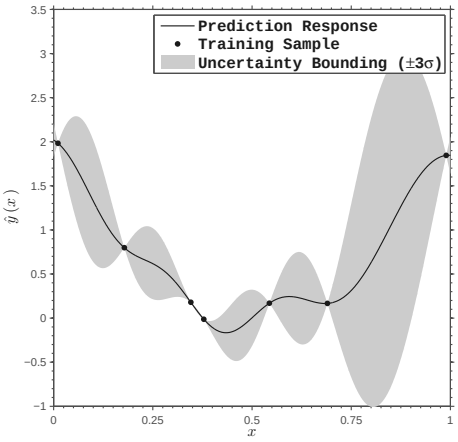


(a) Surrogate Prediction with Uncertainty

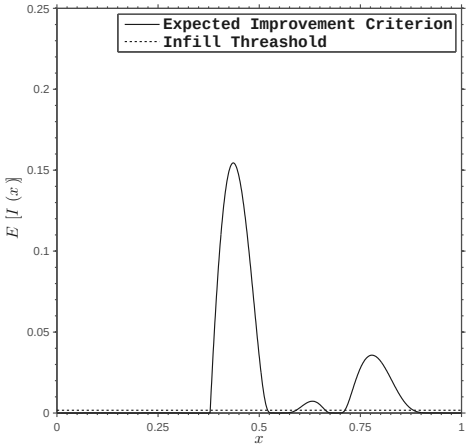


(b) Expected Improvement Response

Figure 2.3: 1D Expected Improvement Iteration 1

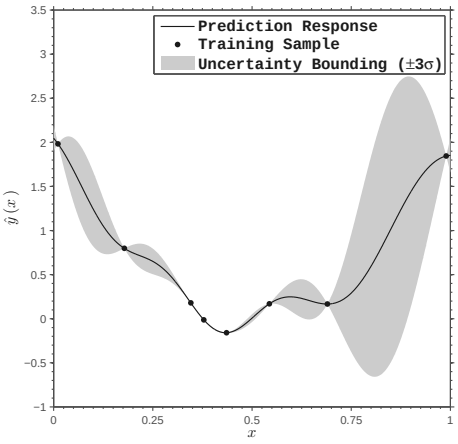


(a) Surrogate Prediction with Uncertainty

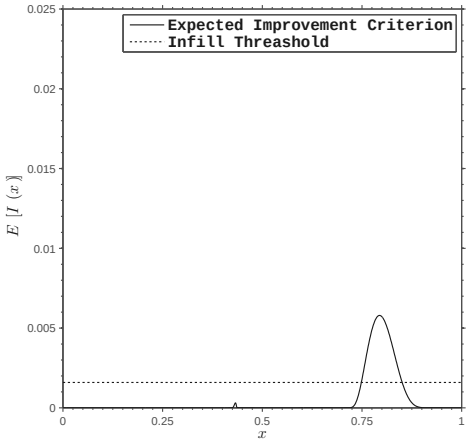


(b) Expected Improvement Response

Figure 2.4: 1D Expected Improvement Iteration 2

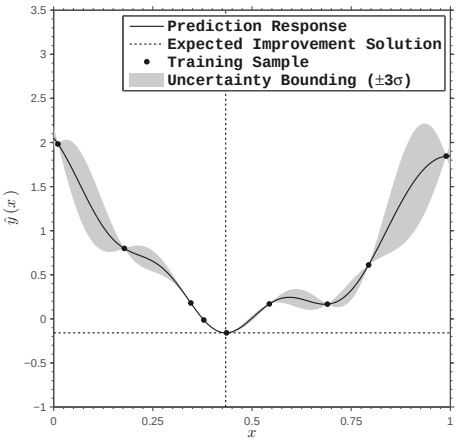


(a) Surrogate Prediction with Uncertainty

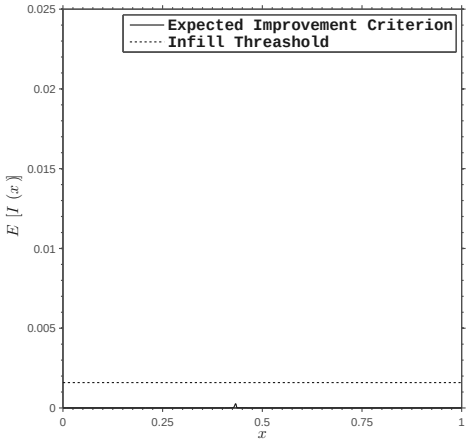


(b) Expected Improvement Response

Figure 2.5: 1D Expected Improvement Iteration 3



(a) Surrogate Prediction with Uncertainty



(b) Expected Improvement Response

Figure 2.6: 1D Expected Improvement Iteration 4

Local Surrogate SORA

Among the first actions taken leading to the development of the proposed methodology was an implementation of SORA which uses local validated surrogates for all analysis responses, referred to as Local Surrogate SORA. The core features of this tool were selected based on deliverables specified by research funding party, Caterpillar Inc. Such requests included the usage of preferred analysis tool, Dakota (Design Analysis Kit for Optimization and Terascale Applications), which is developed and maintained by Sandia National Laboratories. Operations handled by Dakota include design of experiments, surrogate model generation, constrained optimization, and inverse MPP searches. These operations utilizing Dakota, necessary for the driving of the local surrogate SORA algorithm, are prepared, invoked, and post processed using a framework developed in Python. Each constrained optimization and inverse MPP search is performed upon surrogate surfaces validated using cross validated and updated when necessary in the vicinity of the highest cross validation error.

3.1 Analysis Features

Among being a surrogate driven version of SORA utilizing Dakota, there are several notable features included in the local surrogate SORA tool for performing RBDO, some of

which are described in the following sections.

3.1.1 Input File Specification

For the problem definition of an RBDO procedure, information such as variable and parameter uncertainty as well as objective and limit state evaluation must be defined. Furthermore, for local surrogate based analysis, information such as sampling criteria, acceptable prediction error, and surrogate training method must also be supplied. For initialization of such information in local surrogate SORA, an input file parsing module was implemented supporting a clear method for all input arguments. In total, seven groupings of input specification, each identifiable by keyword headers, were created to fully define the surrogate based RBDO procedure. Upon program invocation from a terminal, this input file is passed as the only input argument. In addition to accepting program input from the previously described input file, both a python dictionary and "pickled" python object can be used for this purpose. An example of an input file is presented at the end of this chapter before demonstrating local surrogate SORA.

The first input header, `simulation`, identifies the grouping for general process input specification. Here, the simulation name, output verbosity level, maximum iterations, and design convergence tolerance is specified using secondary keywords. Since the results from local surrogate SORA can be incrementally appended to a file while within a complex subroutine call stack, a certain amount of output stream syncing must take place but can significantly reduce the speed of the overall process. Therefore an additional input argument is specified for the turning off of output file syncing with all results being presented upon process termination.

The second through fourth input headers, `deterministic_design`, `random_design`, and `random_parameter` define each class of system variable and parameter. Secondary

keywords common between all these groups exist for specifying the number of, and descriptors for, the corresponding variables or parameters. Unique to the design class groupings are a means of entering the starting point and side constraints for each variable. Unique to the random class groupings are means of entering the distribution type along with the mean and standard deviation for each random variable or parameter.

The fifth header, `responses` exists for the entering the simulation evaluation information. This includes the overall number of responses implemented and defined in the RBDO formulation, the truth module which contains the callable evaluation functions, and the reliability requirements for each limit state function. Here, descriptive identifiers can also be entered for differentiation between responses in the process output file.

The sixth header specifies the analysis methods within Dakota to be used for the process operations with keyword `dakota_methods`. Algorithms for four types of analyses must be specified, all of which must be included in Dakota. These analyses include the methods for the initial design of computer experiments, surrogate model type, constrained deterministic optimization, and MPP search. The text representation for each Dakota algorithm must appear as it would in the Dakota input file itself.

The seventh and final header keyword, `surrogate`, is used to specify information related to surrogate region validation. The initial search region size of the surrogate and the number of initial samples used to construct surrogates for each operation are entered in this grouping. Since three different surrogate based operations can take place in local surrogate SORA, different initial sample counts for optimization, inverse MPP searches, and reliability assessment can be assigned. Also contained in this grouping is the acceptable cross validation error threshold for surrogate based analysis.

3.1.2 Utilization of Dakota

As previously stated, design of experiments, surrogate training, optimization, and MPP searching is outsourced to Dakota in local surrogate SORA. A wide variety of methods for each of these procedures is available in Dakota, most of which can be utilized using the file input described above. Most heavily tested algorithms include the gradient based methods for optimization and inverse reliability assessment. For a complete description of all the algorithms in Dakota, the reader is referred to the Dakota users manual [1]. It was found that under most circumstances for most optimization algorithms, the optimal results were located with the expected performance characteristics per algorithm. However, in supporting the use of numerous routines, it was observed that expecting successful termination from all requested algorithms, and thereby ignoring search failures, would not provide a robust software package. Most observed failures of Dakota algorithms were due to convergence criteria not being met prior to exceeding the maximum amount of iterations. Even with the most thoughtful convergence and maximum iteration specifications, it was clear a static setting would not work 100% of the time for the many runs necessary for one local surrogate SORA procedure. Therefore, one exhaustive algorithm for both constrained optimization and inverse MPP searches were implemented in local surrogate SORA to be executed when an algorithm failure was detected in the Dakota results. With this addition to local surrogate SORA, the software as a whole becomes much more reliable regarding solutions found.

3.1.3 Local Validated Surrogates

A major feature of local surrogate SORA is the usage of local regions for all optimization and MPP searches, similar to a trust region method for optimization. The location, size, and sampling density of subsequent regions is based on movement of the optimal point and

number of samples required for surrogate validation. To ensure the generation of a local surrogate with prescribed accuracy, a surrogate validation procedure was included in the tool.

The initial size of the approximated region for an analysis procedure is based on a specified percent of the total design or uncertain space. Each regional surface is optimized upon with the requested Dakota analysis method, and when the optimal solution is located at a region bound, the next region is reconstructed centered at the current most optimal point. The next reconstructed region size is governed by the distance between the previous and current best solution point, each of which must have occurred at a bound or been the starting point. The range for each dimension within a region is equal to twice the point movement in the corresponding direction. The interval, half the range of the next region, is expressed as

$$L_{i,int}^{(k+1)} = \left| x_i^{*(k-1)} - x_i^{*(k)} \right| \tag{3.1}$$

with the lower and upper bounds for the next local region expressed respectively as

$$L_{i,lb}^{(k+1)} = x_i^{*(k)} - L_{i,int}^{(k+1)} \tag{3.2}$$

and

$$L_{i,ub}^{(k+1)} = x_i^{*(k)} + L_{i,int}^{(k+1)} \tag{3.3}$$

where $x_i^{*(k-1)}$ is the previous optimum point and $x_i^{*(k)}$ is the current optimum point for the i^{th} dimension. Given this, it can be seen the size of the next region compared to the current region cannot increase in size. At most, it can retain the same size, but only if the current optimum point is located in a corner of the region. If the optimum point is located at one dimensional bound, the interval for the next region will be largest in that particular direction, but the interval will decrease with respect to all other dimensions.

The initial number of truth model samples used for training each surrogate region is dictated by a sampling factor. Using this approach, the goal is to anticipate the number of samples required for a validated surrogate given the size of a new region. The sampling factor is computed as the ratio between the number of samples and the region size. Since new regions are very near previous ones, here the assumption is a similar number of samples per design space volume is required to approximate the surface with the same level of required accuracy.

Validation of each local surrogate is performed using leave one out cross validation. Since a residual amount is computed at each truth model observation, an average prediction error can be computed for the local region or the maximum of these residuals can be used an uncertainty measure. If the maximum residual exceeds the specified error threshold, a two step iterative process is undergone to locate supplemental samples. Starting with the initial design obtained by optimal Latin hypercube sampling, supplemental points are added to the surrogate training points until the bounds for standardized residual suggested by Jones 1998 [21] are satisfied as well as a threshold percent error.

The first step after initial sampling is to determine if supplemental samples are required to improve the standardized residual values. If the maximum value within the set of standardized residuals is outside of [-3.0,3.0], then at least one supplemental point is required. Let the point corresponding to the highest standardized residual be $x^{\mathcal{E},max}$. The nearest $3k$ points to $x^{\mathcal{E},max}$ are located using a sorted in ascending order vector containing the euclidean distance from each sample point to $x^{\mathcal{E},max}$ described by

$$d_e = \text{sort}\left(\left[\left\|x^{\mathcal{E},max} - x^{(1)}\right\|, \left\|x^{\mathcal{E},max} - x^{(2)}\right\|, \ldots, \left\|x^{\mathcal{E},max} - x^{(n)}\right\|\right]\right) \tag{3.4}$$

The first $3k$ points are extracted from d_e and defined as the vicinity points, X_v.

To avoid the risk of inadvertently selecting a supplemental point near an already existing

sample, the angles between all the possible pairs of vectors formed by $x^{\mathcal{E},max}$ to each point in X_v are compared. The list of vicinity vectors is described by

$$V_v = \left[X_v^{(1)} - x^{\mathcal{E},max}, X_v^{(2)} - x^{\mathcal{E},max}, \ldots, X_v^{(3k)} - x^{\mathcal{E},max}\right] \tag{3.5}$$

with the angles for every pair of vicinity vectors is described by

$$\gamma_{i,j} = \left[\arccos\left(\frac{V_v^{(i)}}{\left\|V_v^{(i)}\right\|} \cdot \frac{V_v^{(j)}}{\left\|V_v^{(j)}\right\|}\right)\right]. \tag{3.6}$$

The magnitudes of each vector for any pair of vectors whose angle is less than a specified amount, γ_{min}, is compared with the vicinity point with the larger magnitude being removed from the vicinity list. Generally, $\gamma_{min} = 30^\circ$ is used. All remaining points are referred to as the reduced vicinity points, X_{rv}. The point in X_{rv} which corresponds with the highest standardized residual in that subgroup is selected as $x_{rv}^{\mathcal{E},max}$. The supplemental point is selected as the midpoint between $x^{\mathcal{E},max}$ and $x_{rv}^{\mathcal{E},max}$ as

$$x_{sp} = \frac{x^{\mathcal{E},max} + x_{rv}^{\mathcal{E},max}}{2}. \tag{3.7}$$

Supplemental points are repeatedly added until all standardized residuals are within the bounds [-3.0,3.0]. The process is then repeated once more to decrease the maximum percent error which correspond with the sample points by setting $x^{\mathcal{E},max}$ and $x_{rv}^{\mathcal{E},max}$ to the points which contains the largest amount of percent error rather than standardized residual. The supplemental sampling procedure is terminated only after the largest cross validation percent error is below the specified threshold after the standardized residual bounds have already be satisfied in the first step.

3.2 Implementation

The local surrogate SORA tool was implemented in Python taking full advantage of the object oriented programming paradigm. This section is devoted to providing an overview of the major operations from a development standpoint which make the above features possible. A class hierarchy was designed, utilizing inheritance in appropriate situations to avoid redundant properties and functionality in multiple class definitions.

3.2.1 SORA Process

The highest level procedure taking place in local surrogate SORA is the SORA process itself which invokes all sub-processes. A flowchart showing program flow of the general SORA algorithm is shown in figure 3.1. This main procedure is defined within its own class where all lower level procedures belong to sub-classes which are attributes of the SORA process. It can be seen from figure 3.1 that two sub-processes, optimization and inverse reliability assessment, are referred to as 'core operations'. Classes defining actions taken during these sub-processes of SORA inherit base attributes from a core operation class. Defining characteristics of a core operation are the usage of a local validated surrogate and call of a Dakota algorithm for optimization solutions. It should be noted that one core operation may include multiple local region optimization calls before locating a solution which does not occur at one or more region bounds. The first core operation for deterministic optimization determines the new optimal design given the current shifting vector, providing information to check algorithm termination criteria. The second core operation, inverse MPP search, provides information for the updating of the shifting vector if the design has not yet converged.

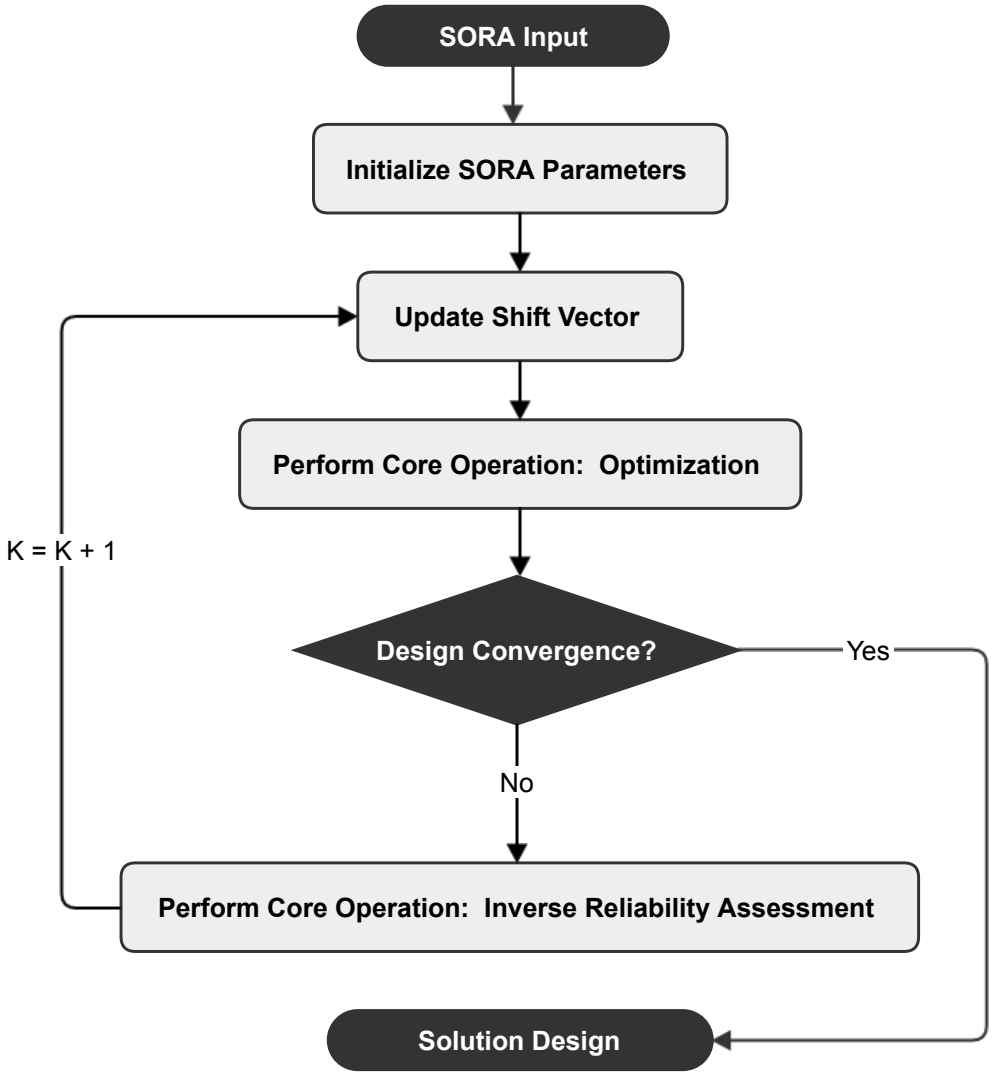


Figure 3.1: SORA Process Flowchart

3.2.2 Core Operation

The second highest level procedures are each of the core operation objects which reside on the SORA process object. The goal of a core operation is to find a search point with certain characteristics within a locally defined surrogate region. As mentioned in the previous section, classes which implement specialized search routines for both general constrained

optimization and inverse MPP searches exist which inherit the overall optimization process common in both procedures from the core operation class. This overall procedure is described in figure 3.2. The core operation is responsible for defining the current local region and updating it after each local point search until the overall solution is identified as well as acting as invoking the generation of the validated surrogate building process as well as the Dakota processes.

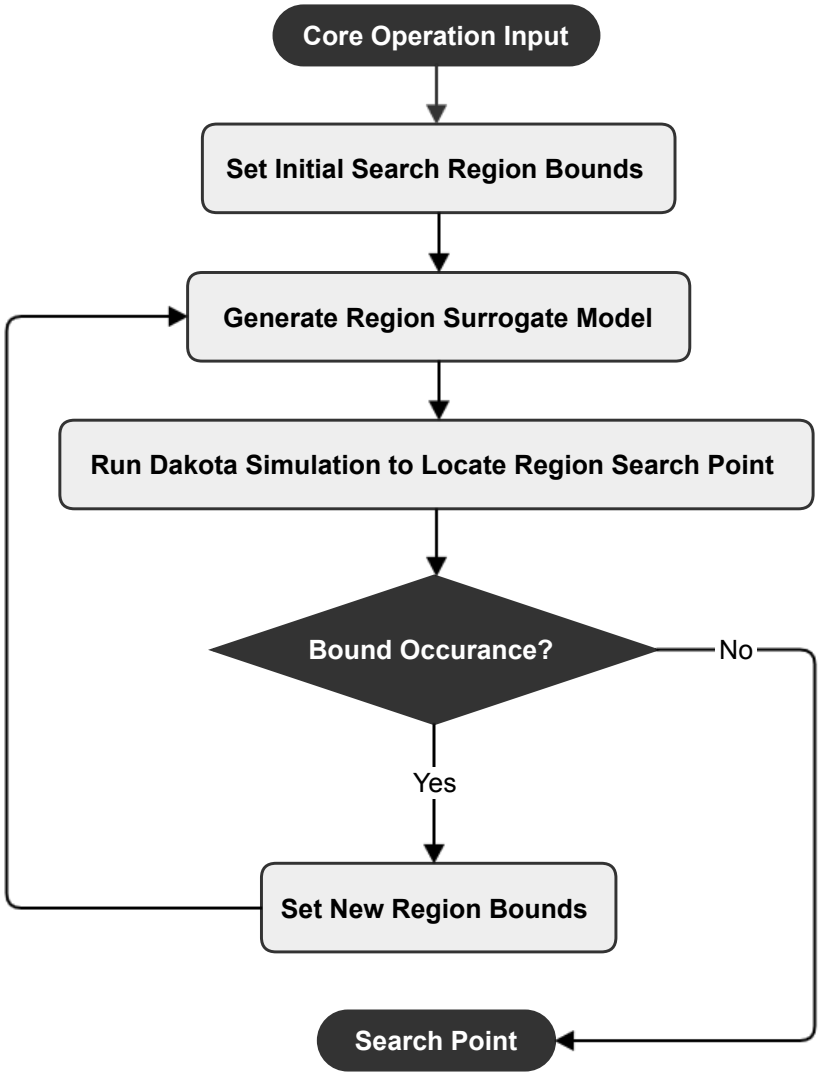


Figure 3.2: Core Operation Flowchart

3.2.3 Surrogate Building

The process for the building of a validated surrogate within a specified region is described in figure 3.3. The class defining functionality for this process can be considered a third highest level optimization since the resulting objects reside upon a core operation object. For this process, since part of the current region may overlap previous regions, all reusable samples are identified and combined with samples from a Dakota LHS for which the number of samples is defined by the sampling factor. With this initial sampling, the resulting surrogates in the cross validation procedure are generated by Dakota with the supplemental sampling procedure invoked when required.

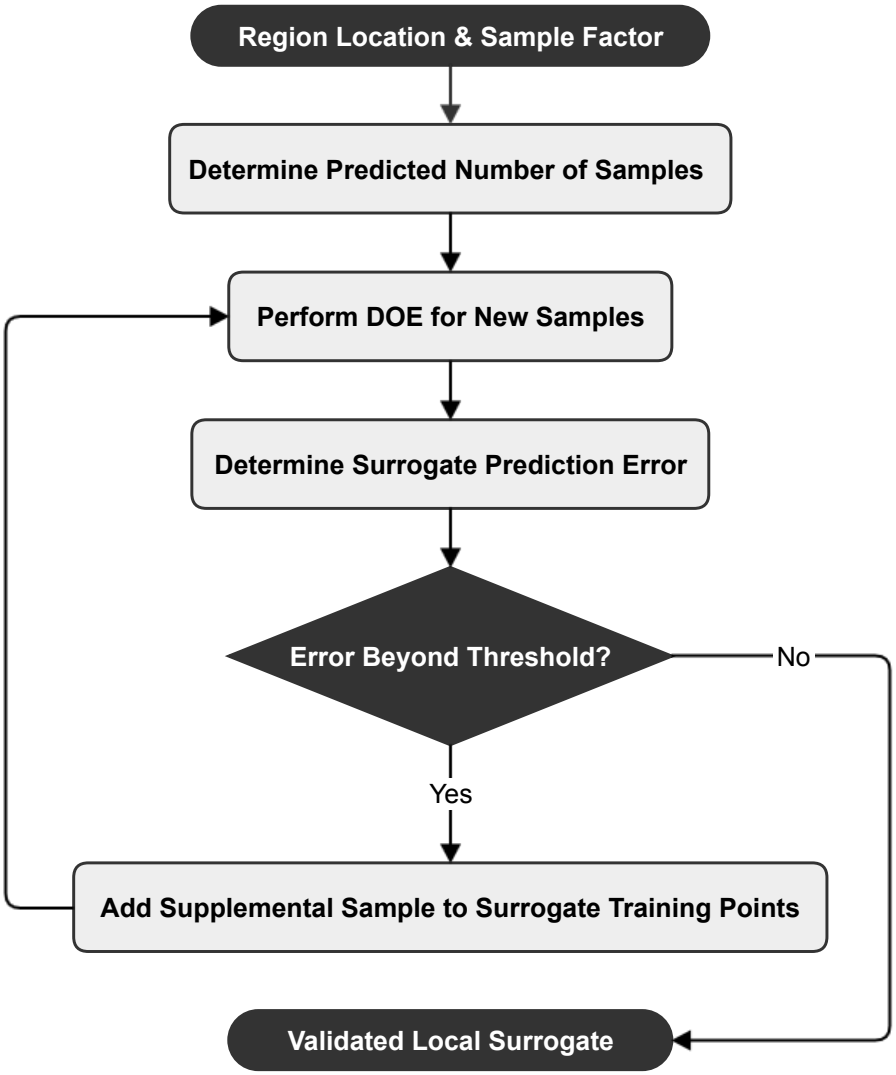


Figure 3.3: Surrogate Builder Flowchart

3.2.4 Dakota Operation

The lowest level, though arguably the most complex procedure included in local surrogate SORA is the Dakota operation class for which the process is described in figure 3.4. This class is also home to two sub-classes, the Dakota input generator and Dakota output parser. Since all input defining a Dakota simulation is text based within an input file, they must

be generated dynamically on the fly for every operation which may be required including optimization, MPP searches, design of experiments, truth model parameter studies, and surrogate model parameter studies. Here, the general process is to create the input file for the requested analysis, invoke Dakota using the input file from the command line - piping the results to an output file, and finally parsing the output file for the results required to continue driving local surrogate SORA.

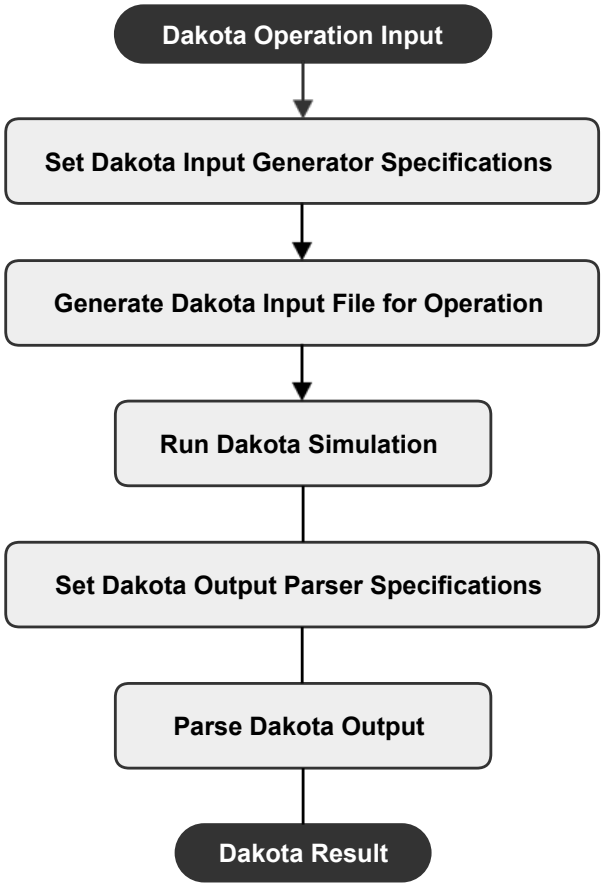


Figure 3.4: Dakota Operation Flowchart

Performing optimization using surrogate responses generated with a known set of truth samples turned out to be a non-trivial task while using Dakota. It was necessary to nest a surrogate parameter study as the analysis driving response function inside of the Dakota optimization routine. This leads to multiple Dakota simulations running simultaneously,

with one waiting for output from another to continue. Due to the complexity required to achieve the described overall process, it is clear the Dakota developers did not intend for their software to essentially be used as a callable library to drive an outside analysis. Dakota includes optimization under uncertainty algorithms which can easily be configured but has yet to include an implementation of SORA. Nevertheless, this software development project provided knowledge and motivation to be applied in developing the main contribution of this thesis, proposed in the next chapter.

Although the described procedure was functionally sound, this method used for surrogate validation was not optimal for the searching of inverse MPPs for use with decoupled RBDO formulations which apply a shifting vector to the deterministic constraints. This is due to the importance of the spacial location of the MPP rather than the limit state response associated with it. Since the sifting vector is computed using the difference between the mean values of the random design variables for the current design and the corresponding inverse MPP dimensions, an inaccurate MPP location will lead to an inaccurate shifting vector causing an unnecessary large number of SORA iterations. The accuracy of the predicted limit state response had been verified within acceptable parameters, but this was not the case for its location accuracy in random space.

3.3 Demonstration

This section is devoted to offering an application example of local surrogate SORA. The focus of this demonstration will be concerned with the usage of local regions with the optimization step in SORA.

3.3.1 Cantilever Example

A closed form cantilever example is used to demonstrate local surrogate SORA. This popular benchmark example contains one object for the minimization of structure weight while upholding reliability requirements against two failure modes involving stress and deflection due to two tip loading conditions. A figure describing the system is shown in figure 3.5.

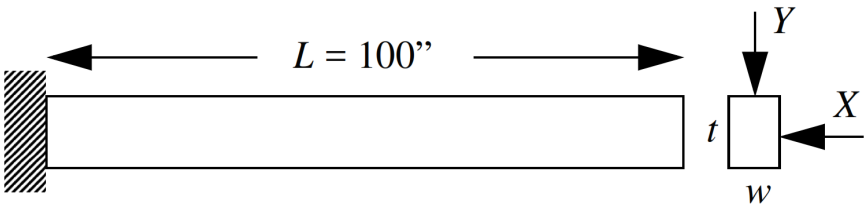


Figure 3.5: Cantilever with Loading

Two deterministic design parameters are defined for the system, the width and thickness of the beam. The objective response for the system under minimization is expressed by

$$f = wt \tag{3.8}$$

where w is the width of the beam and t is the thickness of the beam. Four random parameters are also defined, the yield strength and elastic modulus of the beam as well as horizontal and vertical tip loads. This particular problem does not include any random design variables. The stress limit state is expressed by

$$g_s = \frac{stress}{R} - 1 \tag{3.9}$$

where

$$stress = \left(\frac{600}{wt^2}\right) Y + \left(\frac{600}{w^2 t}\right) X \tag{3.10}$$

for which Y is the transverse tip load with distribution parameters $N\,(1000, 100)$, X is the

axial tip load with distribution parameters $N\left(500, 100\right)$, and R is the yield strength with distribution parameters $N\left(40000, 2000\right)$. The deflection limit stat is expressed by

$$g_d = \frac{displacement}{D_0} - 1 \tag{3.11}$$

where

$$displacement = \frac{4L^3}{Ewt}\sqrt{\left(\frac{Y}{t^2}\right)^2 + \left(\frac{X}{w^2}\right)^2} \tag{3.12}$$

for which L is the length of the beam, E is the elasticity with distribution parameters $N\left(2.9E+7, 1.45E+6\right)$, and D_0 is the allowable tip displacement, set to $D_0 = 2.2535$. Here, the modeling of normally distributed values is $N\left(\mu, \sigma\right)$.

The input file for solving this problem using local surrogate SORA is shown in figure 3.6. Here, it can be seen that the starting location and side constraints for each design variable is set to 2.5 and between 1.0 and 9.0, respectively. The convergence tolerance used for termination of the SORA process is 0.01, which corresponds to the magnitude of the vector representing design point movement from one design to the next. The reliability constraints, entered next to the keyword `betas`, for each limit state defined in the truth driver is set to 3.0, corresponding to a 99.87% probability of success. The initial sample plan used to generate all surrogates is determined by Latin hypercube sampling, as denoted next to the keyword `dace_doe_methods`. All surrogates used are Gaussian process models. Under the `surrogate_methods` header, it can be seen that the initial local surrogate size has been set to ten percent of the total design space and the number of samples selected to approximate the objective response is 20 while the number of samples selected to approximate the limit state responses is 40. Also specified in this section is the acceptable average percent error for each surrogate built, set to one percent for this example.

cantileverRbdoInput.txt

```
simulation
    name cantileverRbdo
    output verbose
    output_synced on
    max_iterations 12
    conv_tolerance 1e-3

deterministic_design
    count 2
    initial 2.5 2.5
    lower_bounds 1.0 1.0
    upper_bounds 9.0 9.0
    descriptors w t

random_parameter
    normal
        count 4
        means 4.0E+4 2.9E+7 500.0 1000.0
        stdevs 2000.0 1.45E+6 100.0 100.0
        descriptors R E X Y

responses
    count 3
    betas 0.0 3.0 3.0
    truth_module cantilever
    descriptors crossArea_obj stress_ls deflection_ls

dakota_methods
    dace_doe lhs
    global_approx_model gaussian_process dakota
    optimization optpp_q_newton
    mpp_search no_approx

surrogate
    initial_space_percent 10
    initial_opt_samples 20
    initial_ims_samples 40
    initial_ras_samples 40
    error_percent 1.0
```

Figure 3.6: Local Surrogate SORA Input File for Cantilever Example

Table 3.1 shows the iterative history of the local surrogate SORA process in solving the cantilever example. Here, it can be seen the width and thickness variables for the cantilever beam converged after six iterations. It is notable however, that the reliability indexes exceed

the specified target amount. This is likely due to the inaccuracies within the constraint response surrogate models even after the cross validation based infilling process. Even a model which tends to generalize well, may provide inverse most probable points which are relatively accurate, but not accurate enough the proper convergence of SORA. The main contribution of this work in the next chapter aims to address the difficulty involved with these searches.

Iter	w	t	$\mathbb{P}(g_s \geq 0)$	$\mathbb{P}(g_d \geq 0)$	β_{g_s}	β_{g_d}
1	2.3709	3.3000	0.55668	0.48285	0.142	-0.042
2	2.4286	3.9119	0.99974	0.99984	3.470	3.612
3	2.4650	3.8549	0.99968	0.99996	3.418	3.993
4	2.4525	3.8749	0.99965	0.99993	3.391	3.826
5	2.4596	3.8636	0.99973	0.99991	3.470	3.755
6	2.4611	3.8614	0.99968	0.99995	3.418	3.924

Table 3.1: Local Surrogate SORA Cantilever Design History

In this problem, three local surrogate regions were used in the first iteration. Figure 3.7 shows the locations of these regions and the optimal point found for each. It should be point out the design space represented in figure 3.7 does not encapsulate the global design space. It has been reduced in the vicinity of the initial starting point to better visualize the described process. Here, it can be seen the starting point was not within the feasible region, however, the small feasible section within the first local region was located with the most optimal point within said region being used as the center point for the second local region. Each iteration shown in table 3.1 uses however many local regions are necessarily to locate the optimum point following constraint shifting. Generally, only a few local regions are necessary for each iteration with thoughtful selection of the initial space percent, and decrease in size as the algorithm continues. By only approximating a subsection of the design space, computation effort can be saved. However, using local regions in this manner introduces the typical risks of locating a locally optimal solution rather than a globally optimal solution.

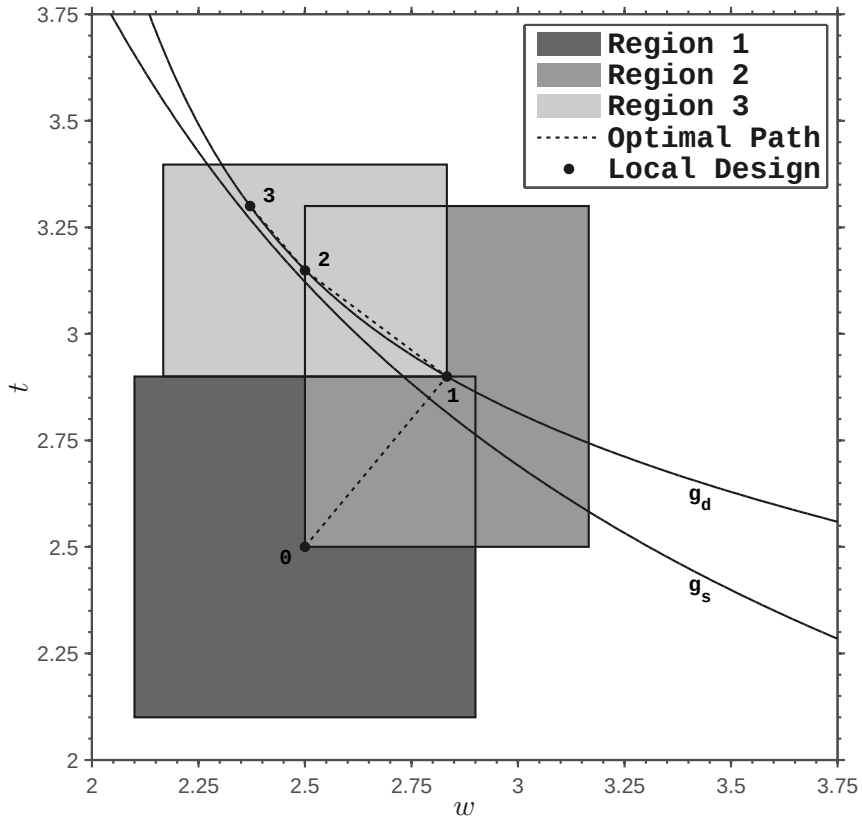


Figure 3.7: Local Design Regions for Iteration 1

Distributed Surrogate PMA

4.1 Literature Review

Many methods have been proposed which aim at defining the most computationally efficient procedure for performing surrogate based analysis. An enduring framework for surrogate based optimization was proposed by Jones (1998) [21] which uses a stochastic process model based upon Kriging for response approximations. Using this approximation model, described by Sacks (1989) [35], yields uncertainly information concerning the predicted value. The search for an optimal point within design space as well as model accuracy improvement is performed simultaneously by globally maximizing a figure of merit referred to as expected improvement (EI). The EI is a function of both the uncertainty of the surrogate prediction and current minimum sample response first described by Mockus (1978) [27]. When a point which maximizes the EI is identified, it is evaluated by the truth model and added to the surrogate training set after which a new EI is found and repeated until the expected improvement is reduced beyond a specified ratio.

Further developments for the usage of EI in surrogate based optimization were proposed by Forrester (2008) [15]. It is pointed out that for multimodal responses being approximated with few samples by Kriging, the optimized model parameters estimated during the likelihood maximization building process may not represent the degree of correlation

present in the true response leading to poor mean square error estimations. Since the EI depends on accurate mean square errors, the infill points determined otherwise could lead to unnecessary iterations. The accuracy of the kriging parameters can be increased by instead maximizing a *conditional ln-likelihood* for a prediction model passing through a goal response at a trial location for EI maximization in a nested optimization formulation. However, the nested formulation for the modified EI increases computational expense to a large degree since the model parameters must be solved for each EI trial point.

Youn (2004) [40] proposed a response surface methodology for PMA based RBDO using a sensitivity enhanced moving least square approximation model. Including the design sensitivities in the model building information is shown to improve the accuracy of results when performing a gradient based analysis but may not be necessary for strictly response based global methods. The sampling sites comprising the design of experiments (DOE) used to build the response approximation are determined using axial star (AS) and selective interaction (SI) sampling. Here, the mean value (MV) first order reliability method is used to identify the iteration region for the SI DOE component. The overall area which the DOE occupies is ultimately determined by a target reliability index and the nonlinearity of the limit state response. Additionally, the hybrid mean value (HMV) method described by Youn (2003) [41] is used for the PMA solution due to its increased robustness.

Apley (2006) [3] discussed the quantification of response uncertainties for surrogate based robust design for which they propose a methodology considering uncertainties from both the probabilistic system parameters and the interpolation surrogate itself using a Bayesian framework resulting in a closed form prediction interval by viewing the true response surface as a random process.

A method for constraint boundary sampling (CBS) was used by Lee (2008) [25] where the figure of sampling merit is proposed using the stochastic properties of a Kriging prediction. Since a higher degree of sampling occurs near the boundaries in this method, the surrogate

will be more accurate along the curve where the MPP is located following reliability assessment using RIA. By using the predicted Kriging response along with its associated mean square error, a probability of feasibility can be computed for all potential sampling locations in the random domain. This probability then weighted according to the normalized distance from the nearest sample in the current surrogate training set to form the sampling criteria.The sequential sampling continues until a termination condition of low relative error and feasibility for the latest found sample is observed in three consecutive iterations.

The prediction error of the surrogate model itself for RBDO was addressed by Kim (2008) [22] where the usage of prediction intervals is suggested for moving least square models. By using upper and lower prediction intervals based on residual response variances and a specified confidence level, a formulation for RBDO which constrains the reliability solutions for an upper and lower reliability level can be constructed. Between the two, the conservative design is ultimately used to account for the uncertainties present in the surrogate due to amount and location of sampling. Additional sampling at the MPP of the most dominant constraint is performed when any probabilistic constraints have not been satisfied, which in this case, means the determined reliability amounts for the upper and lower interval responses are outside of the specified reliability bounds. The prediction interval decreases with every new training sample.

A sequential sampling strategy applicable to forward or inverse reliability assessment based RBDO was outlined by Zhao (2009) [44]. Here, a Kriging model is produced by adding samples at the location where the "bandwidth" of the prediction interval is largest in the neighborhood of the point of interest. Sampling continues until the average of the normalized prediction standard deviation is below a specified threshold. The region for which the surrogate is built is an augmented hyper-sphere in standard normal space where the amount of augmentation beyond the target reliability index is variable in the hope of reusing samples from previous reliability analysis iterations. The change in region size is based on the

magnitude of design change where larger changes reduce the region size.

Lee (2011) [24] proposed performing stochastic sensitivity analysis for sampling based RBDO. Here, all reliability assessments and corresponding sensitives are determined using monte carlo simulations. The Kriging variant used by Lee is dynamic, refereed to as D-Kriging and described by Zhao (2011) [42], in the sense that it selects the most optimal basis function for each new model build rather than specifying which to use for all training sample sets. This optimal polynomial basis function is determined using a genetic algorithm and the optimal correlation parameters, normally denoted as θ, is found using a global pattern search. Also described by Zhao (2011) is a purely exploratory sequential sampling strategy for model improvement in which a new sampling location is found by maximizing the prediction mean square error. In addition to outlining stochastic sensitivity analysis, Lee suggests multiple numerical strategies for sampling based RBDO such as a variable radius hyper-spherical local window for surrogate model generation, a uniform local initial sampling strategy, and feasible constraint filtering.

Dubourg (2011) [14] used subset simulations for the computation of reliabilities and their sensitives where the probabilistic constraint Kriging surrogate errors are propagated to the failure probabilities. Insufficient surrogate accuracy is addressed by a population based adaptive refinement strategy which allows the addition of a set of points to the training samples. This set is identified by first determining a candidate population using a weighted margin of epistemic uncertainty in the form of a density function related to the level of confidence specified. To find the new individual sampling points, a K-means clustering procedure is then applied in order to find the center points of the population groups.

Zhuang (2012) [45] applied expected improvement (EI) as sampling criteria for surrogate based inverse reliability assessment inside a sequential RBDO loop. Following an initial Latin hypercube sampling (LHS), EI is used to simultaneously explore the β-sphere and locate the minimum limit state response using the PMA formulation. For use with multiple

constraints the EI measure is modified to be an expected relative improvement (ERI) so it may provide a valid comparison between functions of different response magnitudes. Furthermore, the optimization to maximize the ERI is performed in polar coordinates to eliminate the β-sphere constraint.

Chen (2014) [10] proposed a means for reliability assessment surrogate accuracy improvement in the vicinity of the current design point using a Local Adaptive Sampling (LAS) region for which to add sample points. The location of the new sample point within the LAS region is that which maximizes either the Constraint Boundary Sampling (CBS) or Mean Square Error (MSE) criterion where the CBS is used if the LAS region contains the limit state constraint boundary. The LAS region radius is formulated to be $1.2 \sim 1.5$ times larger than the target reliability index where the magnitude of the scaling factor is dependent on the non linearity of the limit state response within the β-sphere region. In [9], the same author proposed the use of importance boundary sampling (IBS) for RBDO using Kriging models. Here, the importance coefficients used for selecting sample points are determined using the objective response as well as the joint probability density of the design variables.

Work with using support vector machines (SVM) for modeling decisions functions such as constraints or limit states for reliability analysis was conducted by Basudhar (2008) [5]. An adaptive sampling scheme was presented to identify the location with respect to the design variables, rather than the response, of the decision function boundary. This is accomplished through classifying the feasible and unfeasible regions regardless of the continuity of the simulation responses with the aid of clustering analysis. With the boundaries constructed, the reliability of performance can efficiently be computed using monte carlo simulations (MCS). The adaptive sampling criteria is the prospective distance from the decision boundary itself and any previously sampled points.

Modification of the EGO algorithm for application to the surrogate based reliability as-

sessment problem was done by Bichon (2008) [6] referred to as efficient global reliability analysis (EGRA). Here, the figure of merit for adaptive sampling not only takes into consideration the potential for the true value of the response to violate a constraint, such as the expected violation function introduced by Audet (2000) [4], but additionally, it is concerned with the equality constraint in forward reliability analysis. The merit function, referred to as expected feasibility, returns a large value for points in space which are close to the limit state boundary as well as as having large Kriging prediction uncertainty. The same author in [7] applied EGRA to inverse reliability analysis as well as discussing an approach for computing the confidence intervals for the target probability of failure which the final solution contains.

Choi (2001) [11] proposed a response surface methodology for performance measure approach (PMA) based RBDO using a sensitivity enhanced moving least square (MLS) approximation model. The sampling sites used to build the PMA response approximation are determined using axial star (AS) and selective interaction (SI) sampling with the hybird mean value (HMV) method being used to perform PMA.

Work on increasing the accuracy of a multi-fidelity Kriging model sequentially was done by Gratiet (2012) [17]. Improvement of the Kriging accuracy is achieved by the addition of new samples defined by a criterion computed using leave one out cross validation error terms and the mean square error of the prediction provided by Kriging.

A value based global optimization (VGO) was presented by Moore (2014) [28] in which a sequential sampling strategy is defined for kriging-like surrogates. The approximation model is based on a weighted fit of data from numerous sources of varying fidelity. The sequentially selected points are determined using a value of information criterion. Upon the addition of new samples, the accuracy and expense of each analysis model available is considered using the current prediction responses. The value of information is a measure of the benefit which can be expected from the addition of said information and is meant to trading

between solution quality, potential for improvement, and the cost of further information gathering.

An approach for constraint handling in surrogate based optimization was presented by Parr (2012) [31]. The addition of a penalty term to infill sampling criterion is a popular method of performing constrained optimization with surrogates. This increases the difficulty in the optimization phase of the sampling figure of merit. To alleviate this, it was proposed to use a multi-objective formulation to address objective improvement and constraint satisfaction separately.

The termination criteria for surrogate based optimization was considered by Queipo (2009) [32]. Here it is acknowledged that for the utilization of parallel computing, short cycle SBO is useful in decreasing the overall expense of analysis since the overall number of algorithm iterations becomes the limiting factor. To reduce the number of algorithm iterations, multiple supplement points can be evaluated and added to the surrogate building sample set at a time. Furthermore, it is suggested to assess the merit of additional iterations rather than accepting the current best solution based on target specifications. This is done by calculating the probability of improving the current best solution beyond a certain amount at a given set of supplemental points using the covariance matrix of a Gaussian process as well as the regression trend response at each of the prospective evaluation points.

Viana (2010) [37] proposed the running of the efficient global optimization algorithm with multiple surrogate types within the same iteration. By doing so, multiple supplemental points per cycle may be selected and evaluated in parallel per algorithm cycle. If the prediction mean square error is unavailable within the domain of a particular type of surrogate, the uncertainty information from Kriging is used for the computation of the expected improvement.

Wang (2014) [38] proposed a method for sequential sampling applicable to monte carlo

simulation based reliability assessment using Kriging models. A cumulative confidence level for the computed reliability is presented with supplemental sampling criteria based on the maximization of expected improvement of this cumulative value. The confidence level is computed for each monte carlo trial evaluation, and when averaged, produces a quantity representing the accuracy of the entire estimation. Using the cumulative distribution function, high confidence is computed for points in space returning limit state values near zero and small mean squared errors. Samples are added to the surrogate training set until the reliability converges and the cumulative confidence level is above a certain threshold. This work also proposes a method for sensitivity analysis of the modnte carlo reliability with respect to the random variables without requiring additional monte carlo simulations.

Zhao (2013) [43] proposed the use of a conservative surrogate model for reliability assessment using monte carlo simulations to assure the probabilistic constraints of RBDO are satisfied. Using a conservative design methodologies had been previously proposed such as the constant safety margin approach by Viana (2010) [8] using cross validation error. However, it is argued by Zhao that using a constant margin in this manner produces a surrogate which is overly conservative. Rather, it is suggested to form the conservative margin using the Kriging variance weighted using the relative change in the corrected akaike information information criterion. This method of weighting is shown to reduce fluctuation in the prediction bound while accounting for large amounts of uncertainty present in sparsely sampled regions of the model domain.

4.2 Motivation and Challenges

The efforts involved with the implementation of the Local Surrogate SORA tool for RBDO led to findings which pointed out a fundamental lack of support for sequential surrogate infilling procedures geared toward inverse MPP searches. As the literature review suggests,

there has not been a lack of work in the area of surrogate based optimization infilling, but due to the nature of the shifting vector calculation for decoupled RBDO, the accuracy of the limit state response at the inverse MPP is irrelevant. Only the location of the inverse MPP is of importance. For previous work in the literature, infill criterion is primarily concerned with the accuracy of the objective response and the potential for that response to improve. Surrogate based inverse MPP searches for the task at hand should be primarily concerned with identifying the potential locations of the solution point based on the discovery of important response contours, only increasing the accuracy of responses at extrema locations when doing so will have an impact on solution location.

Another shortcoming which the proposed methodology attempts to address is the arbitrary nature of the termination criterion often specified in adaptive sampling procedures. In using the expected improvement infill criterion, the engineer must decide the termination threshold ratio. Doing this does not offer the assurance of solution confidence with regards to an acceptable uncertainty amount. Selecting a threshold too small may lead to inaccurate results while selecting a threshold too large will encourage redundant information gathering. It has also become common for a static number of infill points to be specified at the beginning of the optimization process if the available computing resources are known for a particular analysis. Neither of these termination methods are driven by the solution uncertainty with respect to the predicted response or the spacial location. If the uncertainty for a solution is quantified, the engineer is able to place meaningful termination criteria upon the adaptive sampling process with real world physical meaning for every degree of freedom for the system.

Computing the uncertainty for a spacial solution based on the uncertainty present within a surrogate prediction is not a trivial task, nor is the selection of infill points to most effectively reduce the solution uncertainty if it is found to be too large. A means of perturbing the surrogate from its mean state based on its continuous variance across the problem do-

main is required. Once a method of perturbation is developed, the approximation error of the uncertainty results based on different numbers of perturbations must be quantified or at least eliminated to a certain degree. Even after an accurate uncertainty approximation has been obtained, it is also unclear how this information can be used to select the next supplemental point. Care must be taken to still simultaneously explore and exploit the surrogate model by narrowing in on promising regions of the surrogate while still providing as much new information as possible to the training process.

4.3 Methodology Approach

The prediction associated with a Gaussian process model represents the most likely response outcome given the truth samples used during training. The surrogate response at every point in the design domain has two properties, a mean and standard deviation which is equal to the square root of the prediction standard error. Therefore, the surrogate can be theorized as containing the random parameters describing a population of infinitely many individual models which follow a correlated Gaussian distribution at every point. The greater the amount of information used to train the surrogate, the fewer the differences between each individual realization following this distribution.

To approximate the uncertainty of a surrogate based solution point, the distribution parameters, mean and standard deviation, must be approximated for every degree of freedom at that point. The standard deviation for one or more degrees of freedom being large could be interpreted as evidence supporting low solution confidence. If there existed a group of solutions which represented the global optimum points for a sufficiently large number of different realizations of the population surrogate, computing the mean and standard deviation along each dimension would be simple. The mean, or centroid, for the group of solutions can be considered the overall solution if the standard deviation for that same

group of solutions is sufficiently low.

With the continual sequential adaptive sampling within design space, the standard deviation for a group of prospective solutions should decrease while converging to a low value assuming only one true optimal location exists. Intermittently however, it should be anticipated the solution standard deviation may increase if and when new potential solution regions are identified. The odds of standard deviation increasement grow with sparsity of the initial sampling plan. Therefore, addition of new information to the surrogate building process through infilling may result in either increasing or decreasing the standard deviation of a group of prospective solutions, but over the course of surrogate based optimization, the standard deviation is expected to decrease.

The infill locations should be selected while taking into consideration the potential solution regions identified which have high densities of solution points. Supplemental sample evaluation in these areas can systematically eliminate possible solution regions until only one remains. Further infilling in the final solution region vicinity will effectively decrease the solution region to a solution point. Rather than only infilling within solution regions, the criterion for this action must include incentive for the discovery of new solution regions. The goals for exploration and exploitation must be maintained in attempting to discover new as well as eliminate solution regions simultaneously.

Following the described approach, a procedure of surrogate based optimization based on the usage of distributed surrogates can be developed. This procedure is outlined in figure 4.1. Each group of "distributed" surrogates is determined using prediction parameters from a "base" surrogate. The base surrogate is simply the initial Gaussian process, but using this terminology will become useful in the content which follows. The group of distributed surrogates are formed after determining a set of virtual samples to be added to the training set of the base surrogate. The optimal solution for each distributed surrogate is solved, yielding a group of distributed solutions which can be used to approximate the uncertainty

of the spacial solution by computing the standard deviation for every degree of freedom, providing values which serve as adaptive sampling termination criteria. The infilling procedure is invoked for situations where the spread in the distributed solutions are too large, in which truth samples are added to the surrogate training set in the vicinity of solution regions at supplemental candidate points which exhibit space filling properties.

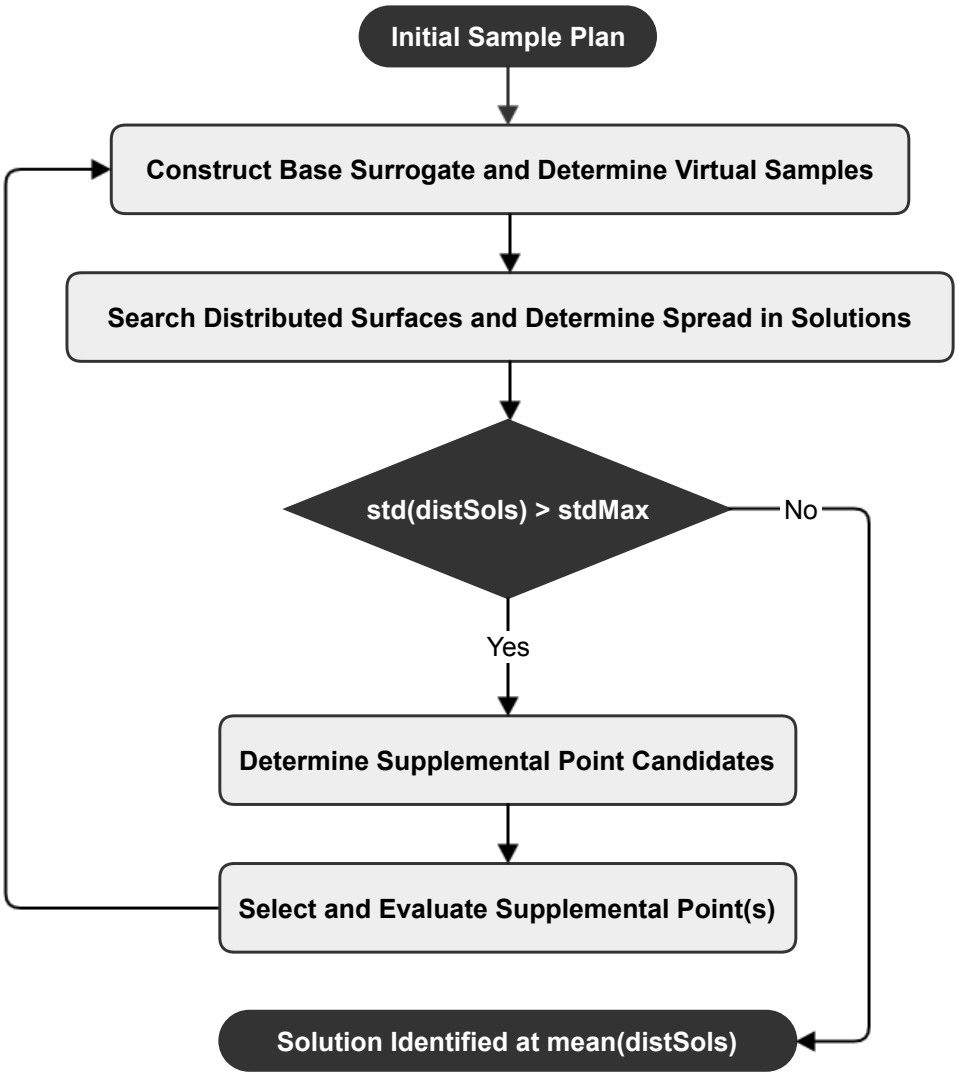


Figure 4.1: Distributed Surrogate Optimization Process Flowchart

4.4 Distributed Surrogate Assessment

The first part of the proposed methodology is concerned with assessing the uncertainty associated with the current best solution within a surrogate based optimization process. This section describes the data driven method used to make this approximation, referred to as distributed surrogate assessment. The theory and reasoning for this process is presented followed by the process specification itself. A convergence study is presented as evidence supporting the criterion for the number of mean surrogate perturbations necessary for solution uncertainty approximation.

The goal of a distributed surrogate assessment is to produce a certain number of distributed solutions from the global optimization of the same number of distributed surrogates and compute the resulting centroid and spread in the distributed solution data. Essentially, this proposes investing computational effort to determine if the probable solutions, given the prediction uncertainty, vary to a large degree and using this in deciding if the adaptive sampling process should be terminated.

4.4.1 Distributed Search

For the formation of each distributed surrogate a certain number of "virtual" samples is added to the base surrogate training set. This results in the mixture of both truth responses and speculative responses based on uncertainties in the predicted surface. Each virtual sample consists of a virtual point representing its location in space and a number of virtual responses based on the mean and standard deviation of the base surrogate at that particular point in space. Since the virtual points must be determined in order to obtain the corresponding base surrogate distribution parameters, their determination is the fist step in searching for the distributed solutions.

Given a sufficiently large specified number of distributed surrogates, the virtual points could be randomly selected within the design space. This approach was initially investigate with some success, but doing so requires the correlation of the virtual responses with respect to the distances between the virtual points, especially if two or more virtual points are selected near each other. Another disadvantage with this method of virtual point selection is the small prediction response standard deviation values for points selected near truth samples which limits base surrogate perturbation. To select the virtual points at the locations of the maximum standard error is a superior method since it maintains good space filling characteristics once the truth and virtual samples are aggregated together for the training of a distributed surrogate, capturing the largest amount of variation in the prediction population. Using this method for virtual point selection, the number of virtual points found is proportional to the number of samples used to train the based surrogate. As the number of training samples increase, so does the number of virtual points since the mean squared error surface will become more multimodal.

For a surrogate model built using five samples, such as the one shown in figure 4.2 for a one degree of freedom system, the virtual points are located as shown in figure 4.3. The prediction MSE surface is optimized upon locally to determine all the points corresponding to local maximum values. By inspecting the locations of the virtual points in figure 4.2 and comparing them to the surrogate building point locations, it can be seen the virtual points occur nearly halfway between build points and the number of virtual samples are one plus the number of build samples for a one dimensional design space.

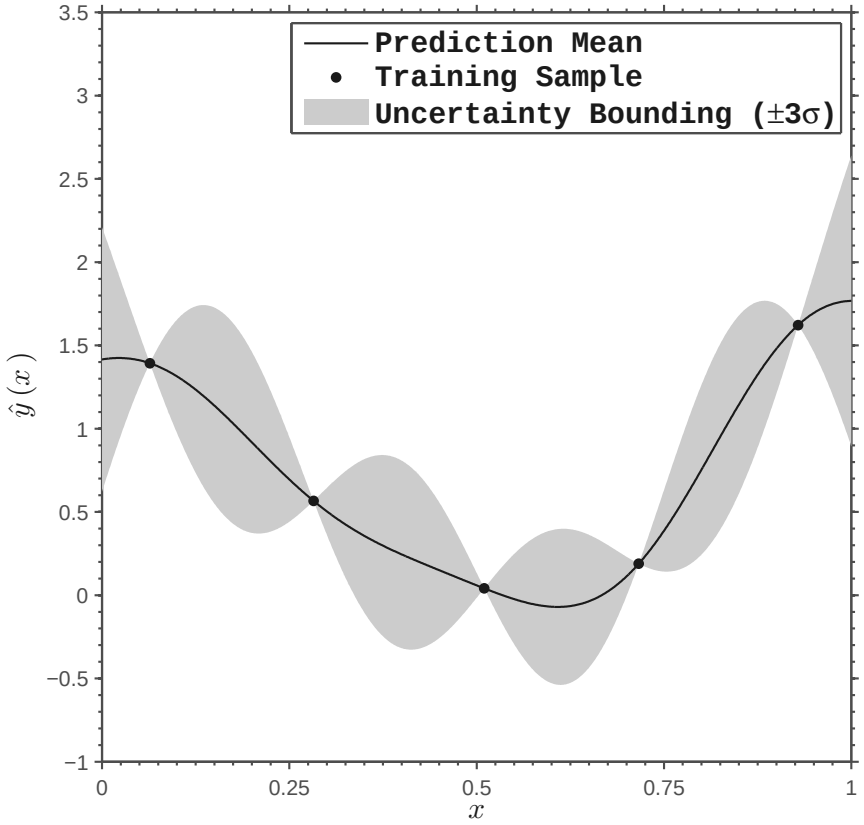


Figure 4.2: Initial Gaussian Process Surrogate

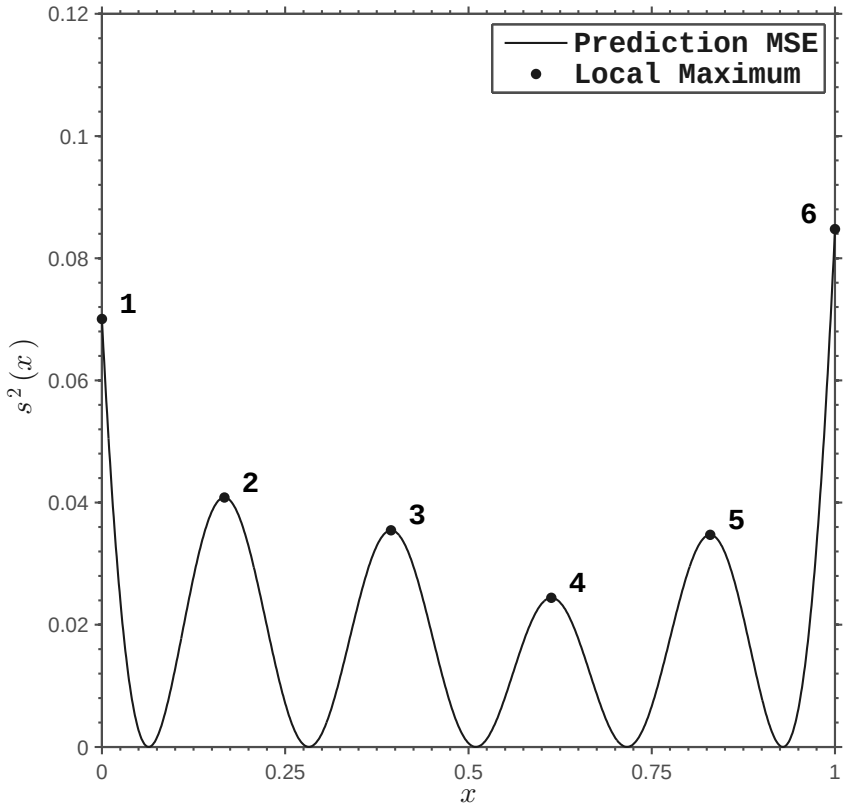


Figure 4.3: Virtual Point Selection

Two pieces of information are extracted from the base surrogate at the location of each virtual point. Both the mean prediction value and standard deviation at each of these points is used to the determine the group of distributed surrogates. For the situation described by figures 4.2 and 4.3, these values are illustrated in 4.1, fully defining the prediction distribution at each virtual point.

	Parameter Values per Virtual Point ID					
Parameter Type	1	2	3	4	5	6
Mean	1.416	1.066	0.256	-0.070	0.989	1.767
Std. Deviation	0.265	0.202	0.188	0.156	0.186	0.291

Table 4.1: Prediction Distribution Parameters at Virtual Points

The virtual responses are selected using a Latin hypercube sample from the corresponding

base surrogate prediction distribution at each virtual point. This provides a means of stratified sampling from the Gaussian distribution associated with each virtual point response prediction as well as a decent combination of one sample from each virtual point location for the defining of a virtual sample set added to the truth samples for one distributed surrogate. If it were computationally feasible, a full factorial design could be generated using all the possible standard score permutations based on a random draw from the prediction distribution, but the resulting number of distributed surrogates could quickly grow beyond a practical number. For instance, for a full factorial design, if there were just five virtual points with nine stratified standard scores, or levels per virtual point, the number of distributed surrogates specified would be equal to five raised to the ninth power - nearly two million. Using the Latin hypercube sampling method is much more convent for representing the uncertainty in the base surrogate with far fewer distributed surrogates, allowing the specification of the number of distributed surrogates desired regardless of the number of virtual points located.

Figure 4.4 describes the design of experiments for nine distributed surrogates using the prediction parameters shown in table 4.1. Here, the grey region centered at each virtual point represents the probability density function for a certain level above or below the mean prediction. The responses for each of the virtual samples is shown in table 4.2. The "Dist ID" in 4.2 refers to each separate distributed surrogate to be generated, short for distributed surrogate identification number. The way in which each response value for a certain virtual point is paired with virtual responses from other virtual points to form a distributed surrogate is the primary reason Latin hypercube sampling is being used. If provided with a list of response levels randomly drawn from the prediction parameters at each virtual point, stratified or not, the mapping of responses to form the distributed surrogate could also be determined by randomly selecting one response level per distributed surrogate from each list corresponding to a virtual point. However, the selected virtual response mapping may not be representative of the random field which the Gaussian process represents if it does

not contain stratification properties, especially for low numbers of distributed surrogates. Thus, it is advantageous to determine the response levels as well as their pairings with responses from other virtual points simultaneously while maintaining the properties of a Latin square in the virtual design space.

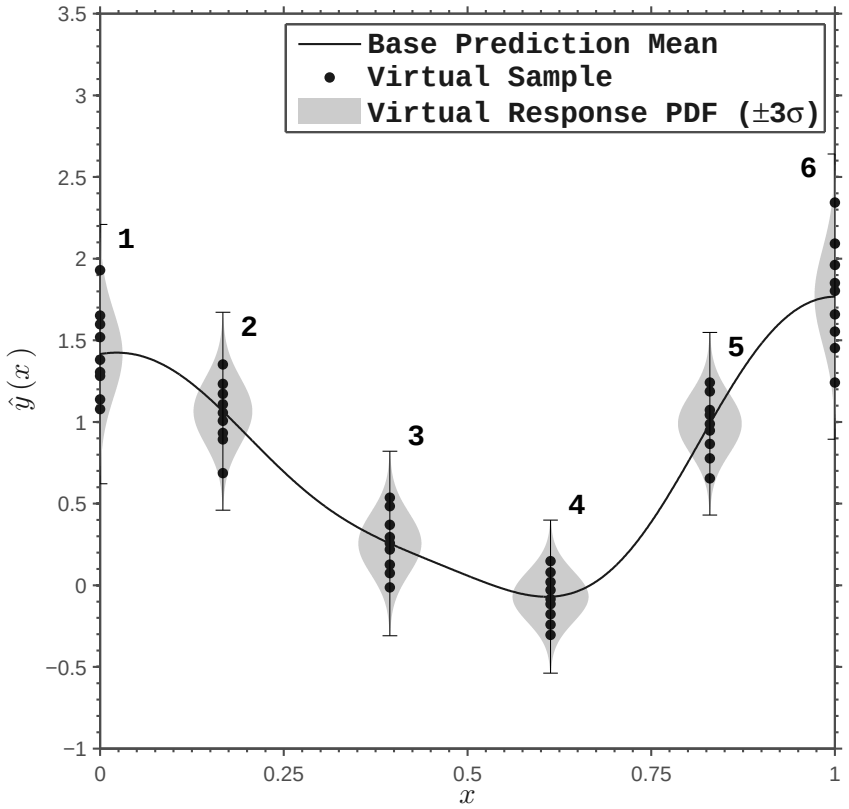
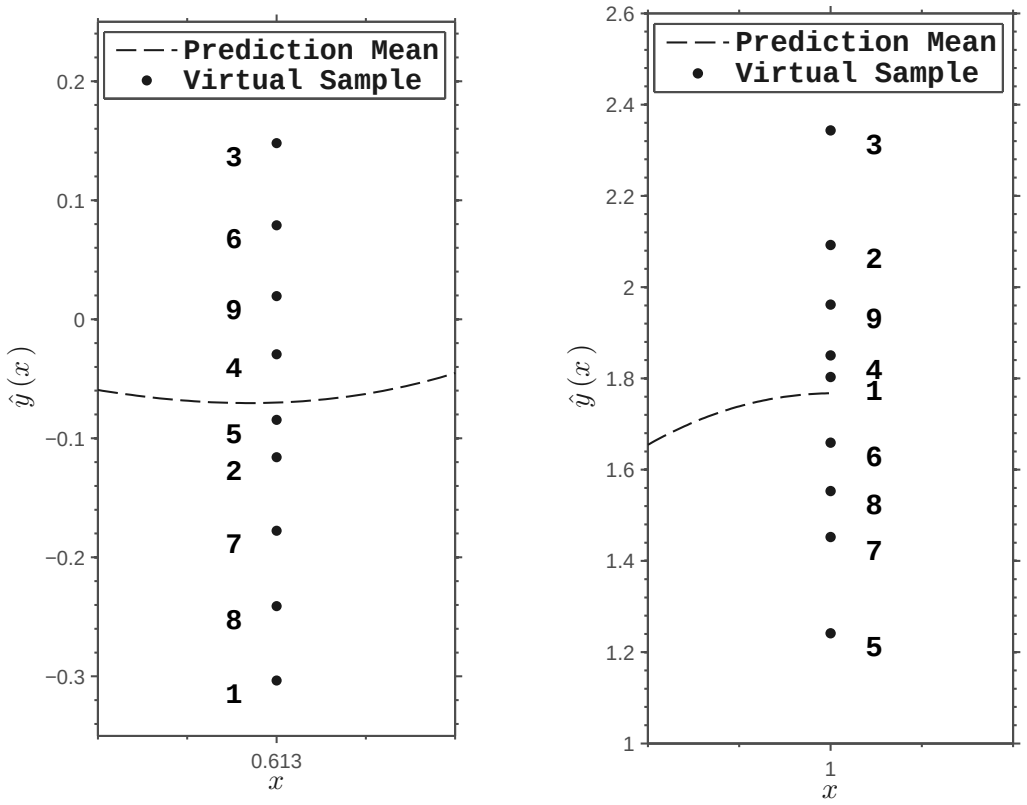


Figure 4.4: Virtual Response Design of Experiments

	Virtual Response Values Per Point ID					
Dist ID	1	2	3	4	5	6
1	1.079	0.686	0.257	-0.303	0.777	1.803
2	1.381	1.007	0.125	-0.115	1.241	2.092
3	1.929	1.352	0.074	0.147	1.043	2.344
4	1.139	1.108	0.218	-0.029	0.987	1.85
5	1.598	0.892	-0.012	-0.084	0.865	1.241
6	1.306	1.233	0.536	0.078	1.073	1.659
7	1.520	1.056	0.369	-0.177	0.947	1.452
8	1.282	0.932	0.484	-0.241	0.654	1.553
9	1.651	1.172	0.294	0.019	1.187	1.962

Table 4.2: Virtual Response Design of Experiments

An enlarged view of the virtual response design of experiments with labels indicating the distributed surrogate identification number is shown in figure 4.5 for the fourth and sixth virtual points to better visualize the described process.



(a) Virtual Point 4 Responses

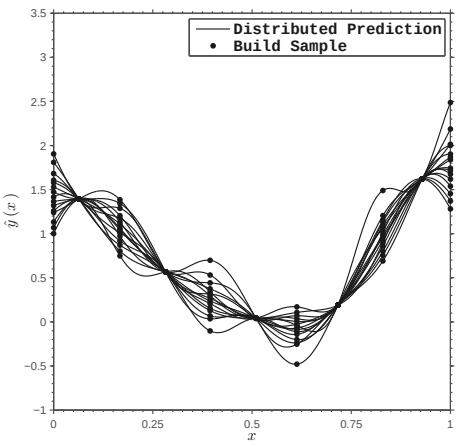
(b) Virtual Point 6 Responses

Figure 4.5: Enlarged Virtual Sample Design of Experiments View

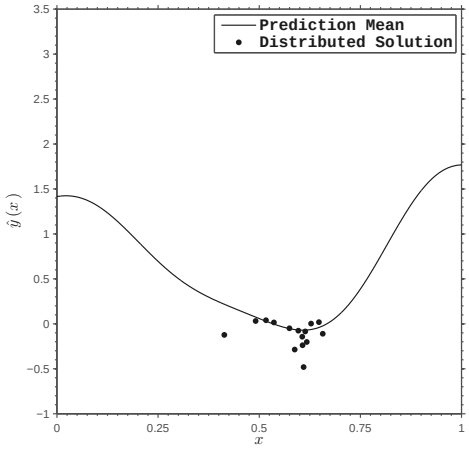
Following the design of experiments for the virtual response sets, each distributed surrogate is trained and globally optimized upon yielding a group distributed solutions. The spread of the solutions with respect to each degree of freedom can be examined and compared to the acceptable real world parameter variation and used termination criterion. The spread in the response values can also be used as termination criterion as well, but generally this criterion is less critical for inverse MPP searches.

To compare the effects of using different numbers of distributed surfaces, the initial Gauassian process shown in figure 4.2 and prediction parameters shown in table 4.1 are used for

three distributed surrogate assessments with said variation. Figures 4.6a, 4.7a, and 4.8a show the distributed surfaces formed using 15, 45, and 135 virtual response levels at each virtual point, respectively. The corresponding distributed solutions, gained by global optimizing on each distributed surface are shown in figures 4.6b, 4.7b, and 4.8b.

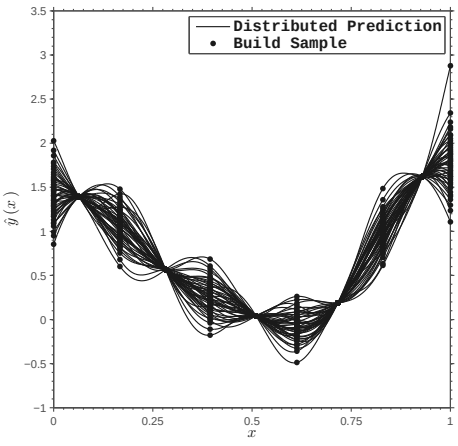


(a) Distributed Surrogate Predictions

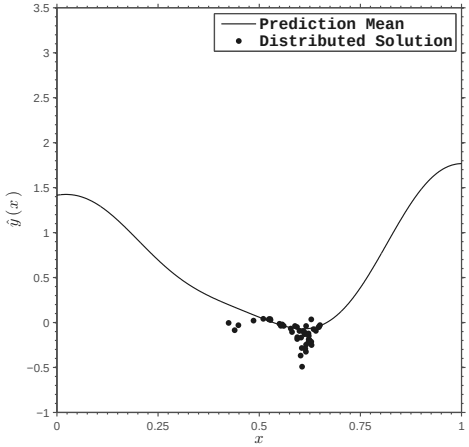


(b) Distributed Surrogate Solutions

Figure 4.6: Distributed Surrogates and Solutions for 15 Surfaces

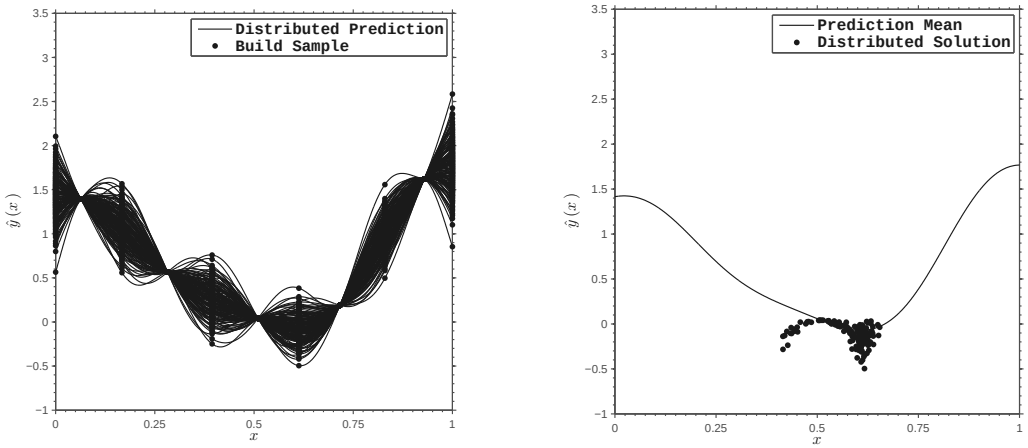


(a) Distributed Surrogate Predictions



(b) Distributed Surrogate Solutions

Figure 4.7: Distributed Surrogates and Solutions for 45 Surfaces



(a) Distributed Surrogate Predictions

(b) Distributed Surrogate Solutions

Figure 4.8: Distributed Surrogates and Solutions for 135 Surfaces

It can be seen the set of distributed solutions gained from each amount of distributed surfaces exhibits a similar pattern. Furthermore, table 4.3 shows the distributed solution mean and standard deviation for the group of points and responses, indicating relative agreement regarding solution uncertainty.

Dist Count	μ_{x^*}	σ_{x^*}	$\mu_{\hat{y}^*}$	$\sigma_{\hat{y}^*}$
15	0.5803	0.0652	-0.1123	0.1432
45	0.5845	0.0559	-0.1108	0.1198
135	0.5766	0.0604	-0.1142	0.1173

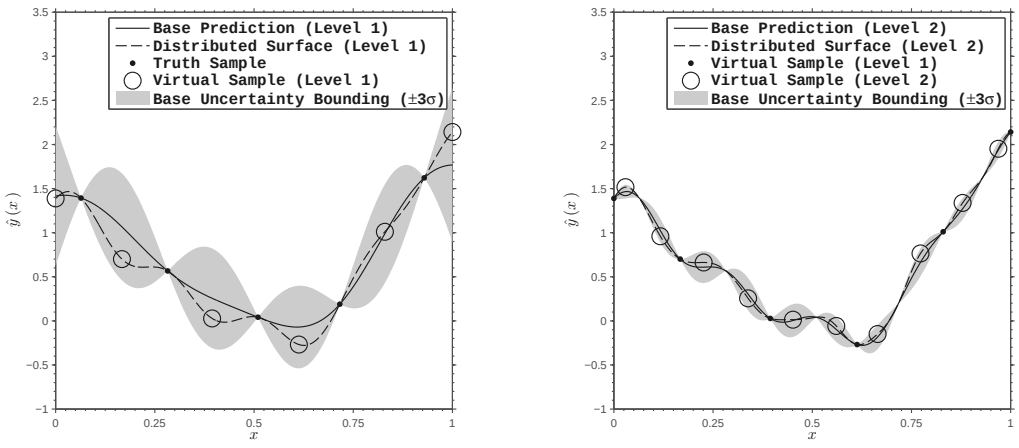
Table 4.3: Distributed Solution Parameters for Different Surface Counts

The standard deviations for the distributed solutions in the x and $\hat{y}$ directions can be compared with the uncertainty threshold specified for surrogate based optimization. For standard deviations exceeding these settings, distributed surrogate infilling will take place under the proposed distributed surrogate optimization procedure.

4.4.2 Multi-Level Assessment

The usage of a discrete set of distributed surfaces in determining the spacial uncertainty of a surrogate based solution may provide approximations which are deceptively low. Since the method of surface perturbation described in the previous section selects virtual responses at a limited number of points, the shape of the perturbed surface is also limited. This hinders the adaptability of each perturbed surface to capture the potential underlying truth behavior. This is most easily recognizable by observing the single modal shape of each distributed surface between truth responses. A more realistic perturbed response could be formed by using a set of virtual samples whose responses are correlated with each other based on the distance between them in regions void of truth samples.

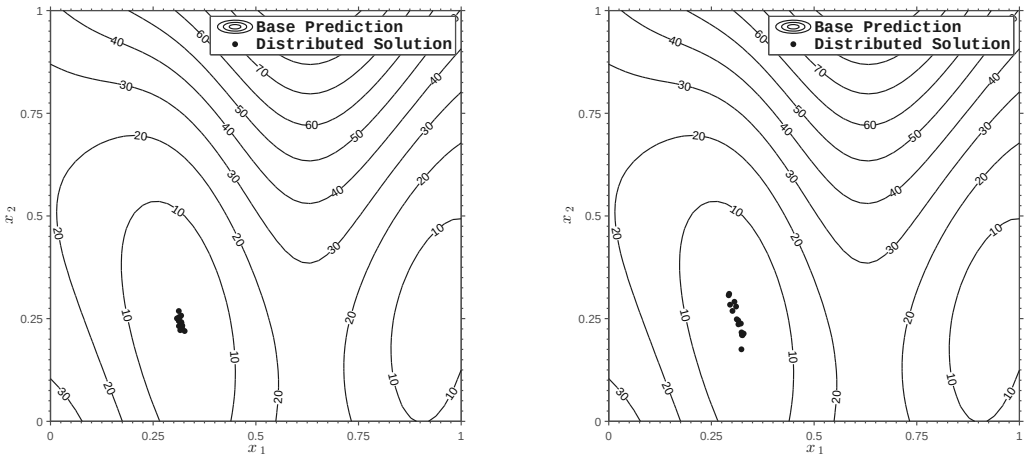
Using a nested approach for the searching of virtual points and their selection of responses is a superior method for producing a set of distributed surfaces regarding solution uncertainty accuracy. Since every distributed surrogate is built upon a base surrogate, there is no reason prohibiting the use of each resulting distributed surrogate as the base for a second level of distributed surfaces. The formation of the second level distributed surface is described in figure 4.9.



(a) Level 1 Surface (b) Level 2 Surface

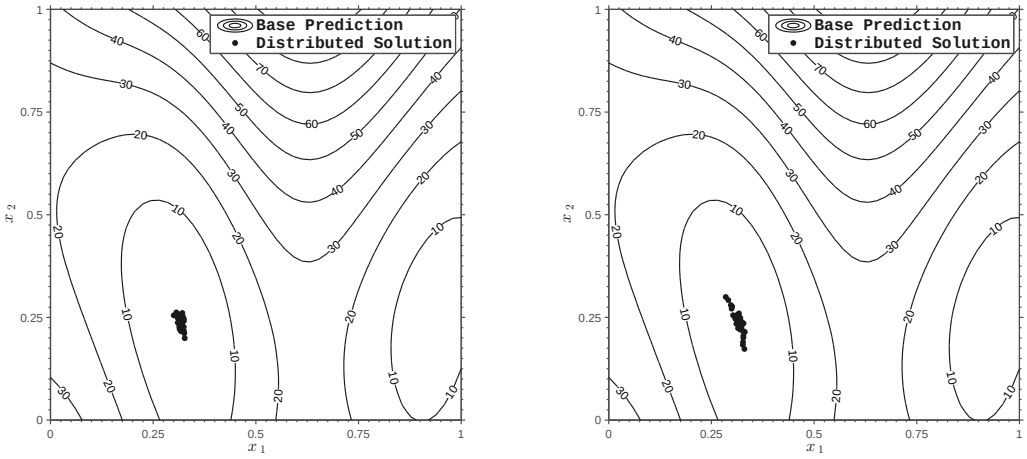
Figure 4.9: Two Level Distributed Surrogate Formation

For the comparison of distributed solutions found using one and two levels, the results of a two dimensional example of such assessments is presented. Since a discrete number of distributed surrogates is selected for the inner and outer distributed surrogates, the product of the two quantities is the amount of total distributed surrogates formed. For a consistent comparison with the single level distributed surrogate counts of 15, 45, and 135, two level distributed count sets are specified as $Set = [5\ 3]$, $Set = [9\ 5]$, and $Set = [15\ 9]$ to produce the distributed surrogates in figures 4.10b, 4.11b, and 4.12b.



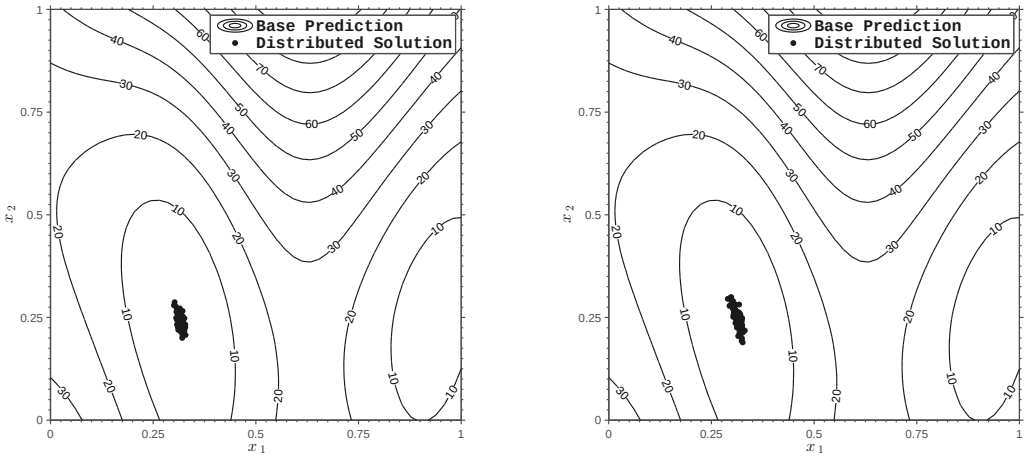
(a) One Level, $Set = [15]$ (b) Two Level, $Set = [5\ 3]$

Figure 4.10: Multi-Level 15 Distributed Solutions Comparison



(a) One Level, $Set = [45]$ (b) Two Level, $Set = [9\ 5]$

Figure 4.11: Multi-Level 45 Distributed Solutions Comparison



(a) One Level, $Set = [135]$ (b) Two Level, $Set = [15\ 9]$

Figure 4.12: Multi-Level 135 Distributed Solutions Comparison

The standard deviation values found for each distributed solution set using one and two level assessments are shown in tables 4.4 and 4.5. Here, it can be seen that the spread in the solutions is generally larger for the two level assessments. Since the two level assessments are assumed to be more accurate for reasons discussed, a two level assessment is incorporated into the procedure for the assessment results used for the numerical results presented at the end of this chapter. However, since two level assessments are more computational expensive than one level assessments, two stages of the algorithm are performed. The first stage involves running of the distributed surrogate optimization algorithm with a one level set of distributed surfaces until the solution uncertainty converges to a small amount. The second stage then uses a two level set of distributed surrogates for the remainder of the process. This generally reduces the time required to perform surrogate based optimization with little effect on final results compared to using two level assessments for the duration of the entire optimization process.

Distributed Set	Distributed Count	$\sigma_{x_1^*}$	$\sigma_{x_2^*}$	$\sigma_{\hat{y}^*}$
$[15]$	15	0.0048	0.0141	0.4067
$[45]$	45	0.0056	0.0146	0.3657
$[135]$	135	0.0056	0.0156	0.4002

Table 4.4: One Level Assessment Spread in Solutions

Distributed Set	Distributed Count	$\sigma_{x_1^*}$	$\sigma_{x_2^*}$	$\sigma_{\hat{y}^*}$
$[5\ 3]$	15	0.0124	0.0401	0.3007
$[9\ 5]$	45	0.0100	0.0253	0.4936
$[15\ 9]$	135	0.0082	0.0225	0.5088

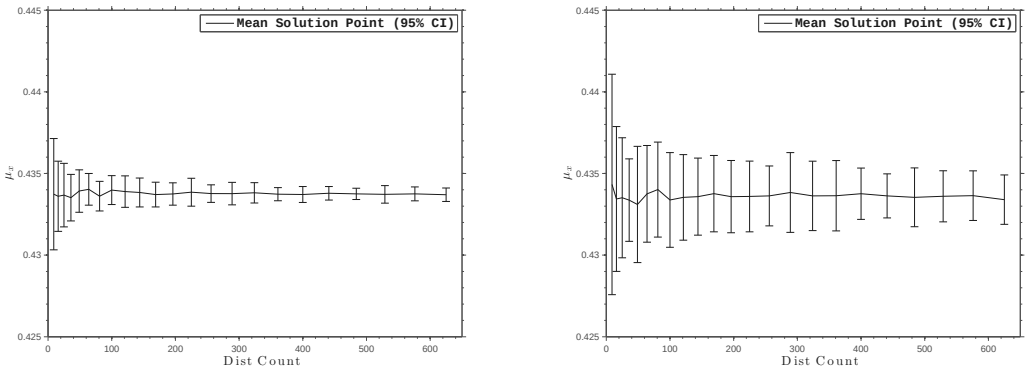
Table 4.5: Two Level Assessment Spread in Solutions

4.4.3 Uncertainty Convergence Study

It is important to acknowledge the training of multiple distributed surrogates does not have computational expense which can be neglected. Each full training of a Kriging surrogate will involve the optimization of the training parameters as well as linear algebra expense depending on the number of degrees of freedom and the number of points used to build it. Therefore, a number of distributed surrogates must be selected which provides an accurate enough representation of the solution uncertainty.

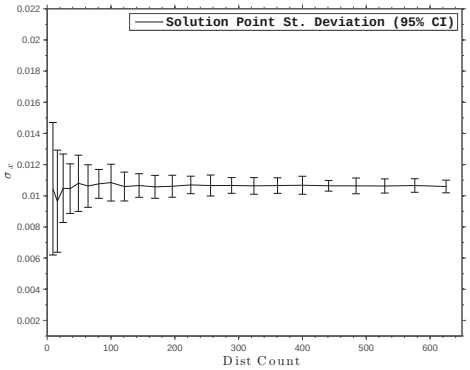
The distributed surrogate assessment can be replicated for the same distributed surrogates for the computation of a mean solution uncertainty and repeated with increasing number of distributed surrogates until the termination the solution standard deviations for all degrees of freedom converge. To show the effect of replicating distributed surrogate assessments in this manner, a convergence study is conducted. Using the 1D example problem as a test case, 25 assessment repetitions are conducted for distributed surrogate counts ranging from 9 to 625 built using both one and two level surfaces. The import results gained by these computational experiments are the mean and standard deviation of the distributed solution

mean and standard deviation from each individual assessment. Figures 4.13, 4.14, 4.15, and 4.16 show the converge of the solution point mean, point standard deviation, response mean, and response standard deviation, respectively. These results confirm that as the number of distributed surrogates increase, the both the spacial solution centroid and spread in the solutions converge. In addition, the general trend of the confidence interval for each mean quantity presented decreases as the number of distributed surfaces increases. This is expected behavior since with higher amounts of distributed surrogates, more space within the virtual design of experiments space is filled. As this space is saturated with a greater number of points, the difference between multiple uncertainty and centroid approximations decrease. To increase the robustness of the distributed surrogate assessment procedure for the numerical results shown at the end of this chapter, the mean of five assessment is generally used for comparison with the solution uncertainty termination criteria. In addition, the number of distributed surfaces used in increased until the solution uncertainty converges up to a predetermined maximum distributed set.

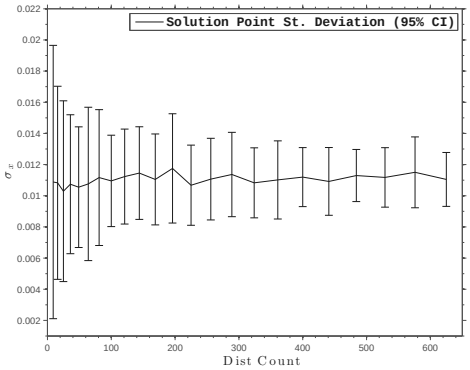


(a) One Distributed Level (b) Two Distributed Levels

Figure 4.13: Solution Mean Point Convergence for 1D Assessment

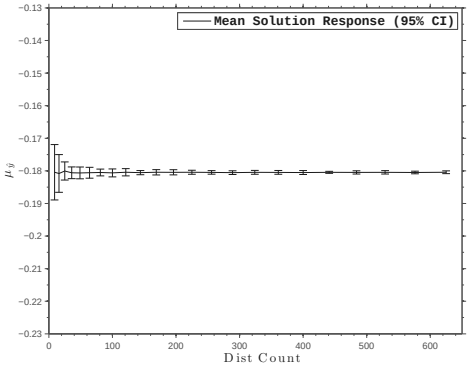


(a) One Distributed Level

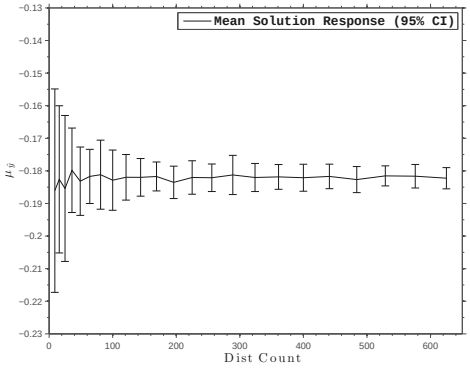


(b) Two Distributed Levels

Figure 4.14: Solution St. Deviation Point Convergence for 1D Assessment

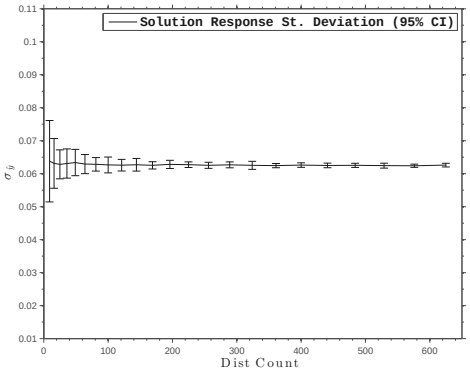


(a) One Distributed Level

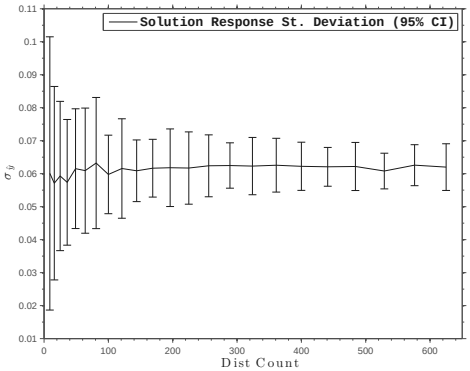


(b) Two Distributed Levels

Figure 4.15: Solution Mean Response Convergence for 1D Assessment



(a) One Distributed Level



(b) Two Distributed Levels

Figure 4.16: Solution St. Deviation Response Convergence for 1D Assessment

4.5 Distributed Surrogate Infilling

Since computational effort is put forth in determining the distributed solutions during the distributed surrogate assessment, it would be beneficial to use this data for the selection of surrogate infills. Since regions which contain the best odds of harboring the true optimal point should be more densely population with distributed solutions, it makes intuitive sense that these vicinities should be under consideration for infilling. However, to avoid evaluation of new samples near preexisting ones, criteria which enforces an exploratory element should be included. The proposed method of infilling determines supplemental candidates, of which all have high exploratory value. The distributed solutions and candidate locations are then compared for determination of infill points.

4.5.1 Supplemental Candidates

Candidates which can be selected after comparison to distributed solution locations must be generated to provide an exploratory mechanism within the algorithm. These candidates are found by identifying all the local maximum mean square error points for the prediction using the same method as the virtual points search during the distributed surrogate assessment stage. In fact, under certain circumstances no additional candidate need to be found, they can simply be assigned to the previously found virtual point set. This is the case for one dimensional problems such as the example used to illustrate distributed surrogate assessment procedure in the previous section. The initial Gaussian process trained using five initial samples shown in figure 4.2 provided the virtual points shown in 4.3. Similarly, the supplemental candidate points for this same surrogate are shown in 4.17. Here, each supplemental candidate is represented as a vertical dotted line.

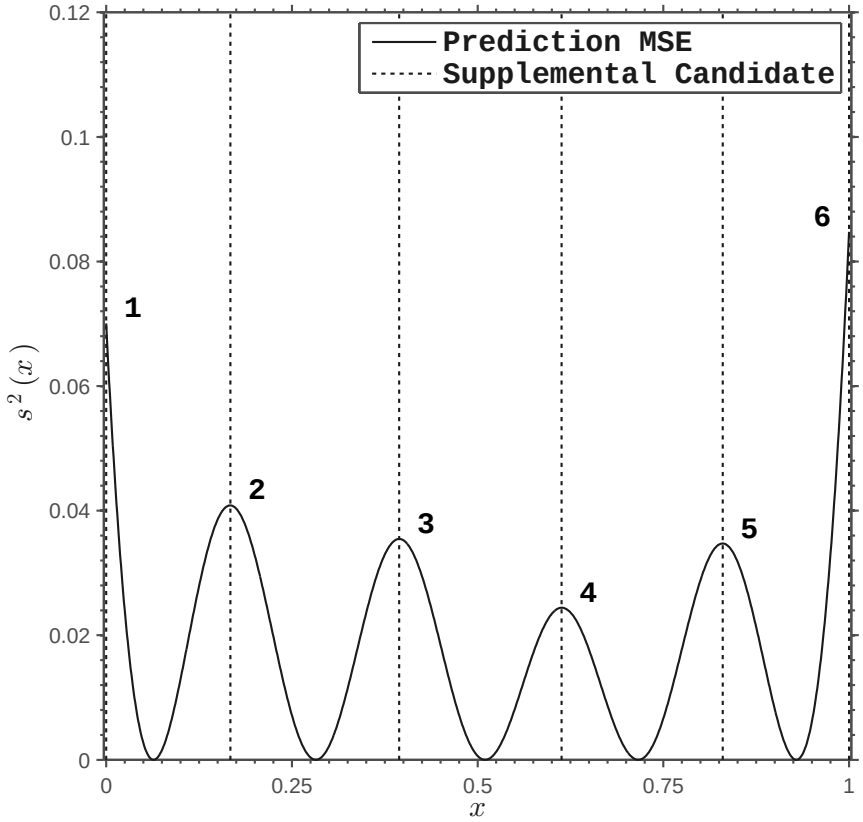


Figure 4.17: Supplemental Candidates

4.5.2 Infill Selection

With the distributed solutions and supplemental candidates located, a subset of the candidate points can be selected as infill points. The distance between each distributed solution and each supplemental candidate is computed with the nearest candidate being selected as a infill point. Using this selection method, the number of infill points selected in a single iteration is only bounded by the number of candidates or distributed solutions, whichever is fewer. Normally however, numerous distributed solutions should be nearest to just a few supplemental candidates unless the initial sampling plan was not appropriate for the dimensionality of the problem regarding the number of truth points, leading to distributed solutions throughout the design space.

Figure 4.18 describes the infill point selection criteria for the distributed solutions shown in figure 4.7b. Here it can be seen two of the six supplemental candidates occur closest to at least one of the distributed solutions. Therefore, the infilling procedure selects the candidates marked by three and four to be evaluated by the truth model for surrogate improvement.

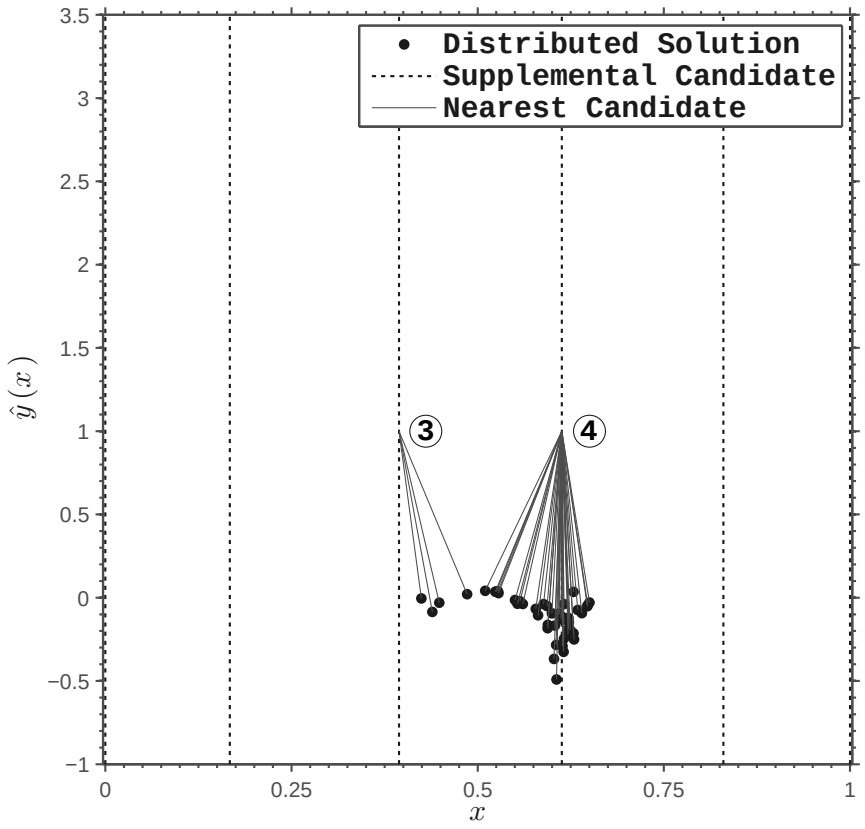


Figure 4.18: Distributed Infill Selection Points

If a limit must be placed on the number of infill evaluations per iteration, a clustering algorithm can be applied to the distributed solutions. Using this method, the distances between the cluster centroids and candidates are compared instead of the distributed solutions themselves. Here, the density based clustering algorithm described by Daszykowski (2001) [12], called Density-based spatial clustering of applicates with noise (DBSCAN) is an attractive method for clustering the distributed solutions. Density based clustering is advantageous for this application because it does not require the number of clusters as an input. Rather,

the two inputs into DBSCAN are the minimum number of points per cluster and the clustering radius. With these two pieces of information the clustering algorithm identifies the natural patterns in the data. Either the radius of the number of minimum points per cluster can be increased sequentially to decrease the number of total clusters identified until the infill limit has been satisfied. This reduction of the distributed solution data leads to an infilling procedure which is less exploratory overall since it is likely to lessen the number of infill point per iteration, possibly increasing of total number of algorithm iterations necessary to reduce the solution uncertainty to an acceptable amount.

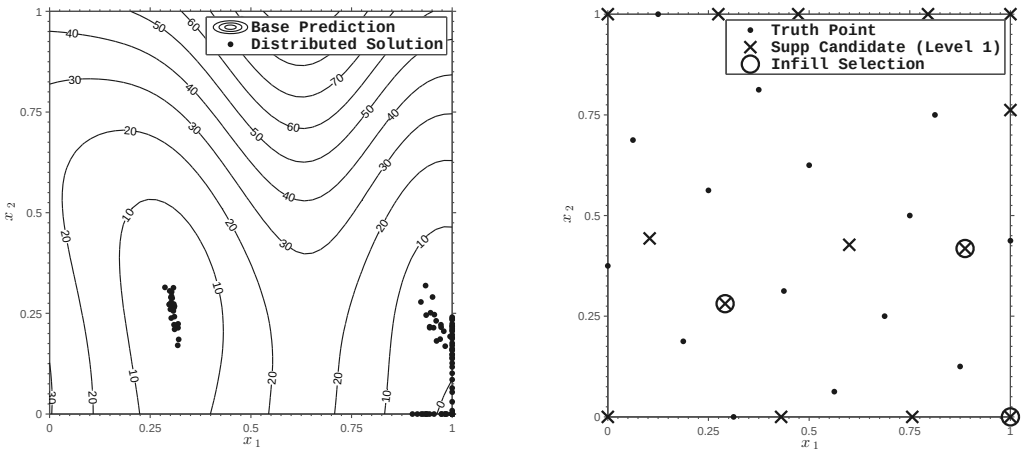
4.5.3 Infilling Heuristics

As the distributed surrogate optimization algorithm continues infilling the design space, the supplemental candidates from each successive iteration should become increasingly concentrated around any remain regions for which distributed solutions continue to appear. This is a natural consequence of using the locations of local maximum prediction MSE as supplemental candidates since every point added to the training set will result in an MSE surface which contains a larger number of local maximums. Furthermore, since infills are added in the vicinity of distributed solution groups, it is intuitive that the new local max MSE locations will appear even closer to the distributed solutions with every new iteration. This phenomena is especially clear when examining the iterative history of one dimensional problems when each candidate selected as an infill is replaced by two candidates flanking its location in the next iteration. The concentration of supplemental candidates around distributed solutions is required for continual solution uncertainty reduction.

With problems having more than one degree of freedom, the iterative concentration of supplemental candidates is slower to develop, and in some cases fails to develop altogether. This results in the infilling of locations around the perimeter of a group of distributed

solutions rather than the center by evaluating points which are located further from the distributed solutions than in previous iterations which do not reduce the uncertainty of the solution. To eliminate this issue, the number of supplemental candidate levels can be increased as the algorithm continues to iterate. This involves the aggregation of the supplemental candidate points found by determining the local max MSE locations of the Gaussian process with a second level of candidate points, also at local max MSE locations for a model which has been trained using the truth samples as well as samples located at the supplemental points whose response is the mean of the Gaussian process.

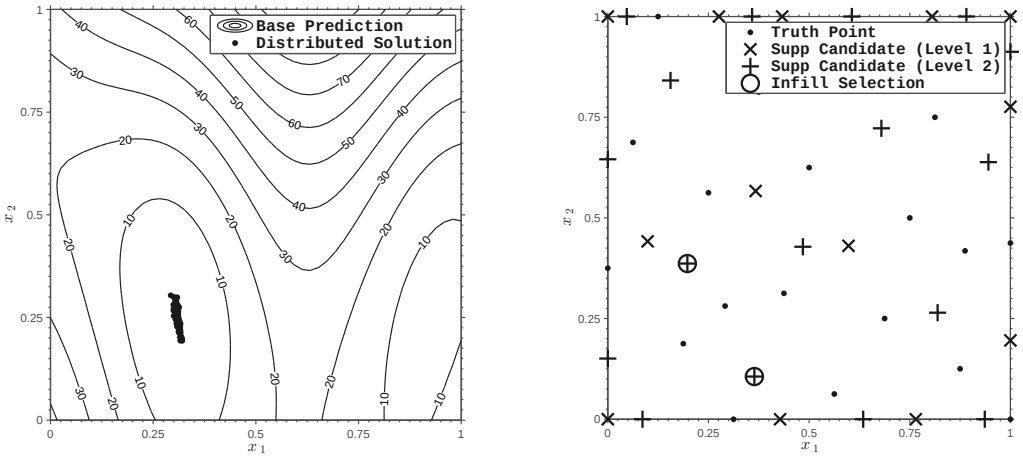
Figures 4.19 4.20 show the infilling process for the first and second iteration, respectively, of distributed surrogate optimization for a 2D problem. Here, two candidate levels have been used to locate the supplemental candidates. It is observed that the number of and density of supplemental candidates increases, making it possible to select infills which are closer to solution regions as the algorithm continues. For all numerical results presented, the maximum number of candidate levels is set to the number of degrees of freedom.



(a) Distributed Solutions

(b) Infill Selection

Figure 4.19: Iteration 1 Infilling of 2D Design Space



(a) Distributed Solutions

(b) Infill Selection

Figure 4.20: Iteration 2 Infilling of 2D Design Space

4.6 Numerical Results

Example problem results using the proposed distributed surrogate method for both global optimization and inverse most probable point searching are presented for one and two degrees of freedom. These results are compared with solutions obtained using Expected Improvement. For one dimensional problems, 90 different initial sample plans generated using Latin hypercube sampling are used. These plans include three subsets of thirty plans, each having either 6, 7, or 8 samples. Similarly, for two dimensional problems, distributed surrogate optimization and expected improvement is conducted using forty different initial sample plans with four subsets of ten plans, each containing 17, 19, 21, or 23 initial samples. These 40 initial sample plans are generated using optimal Latin hypercube sampling. Statistics are shown for each batch of results using the two surrogate based optimization methods. The standard deviation termination criteria for the distributed surrogate method are selected to be 1% of the range of each degree of freedom and expected response range. The distributed set products used in both of the two assessment levels used

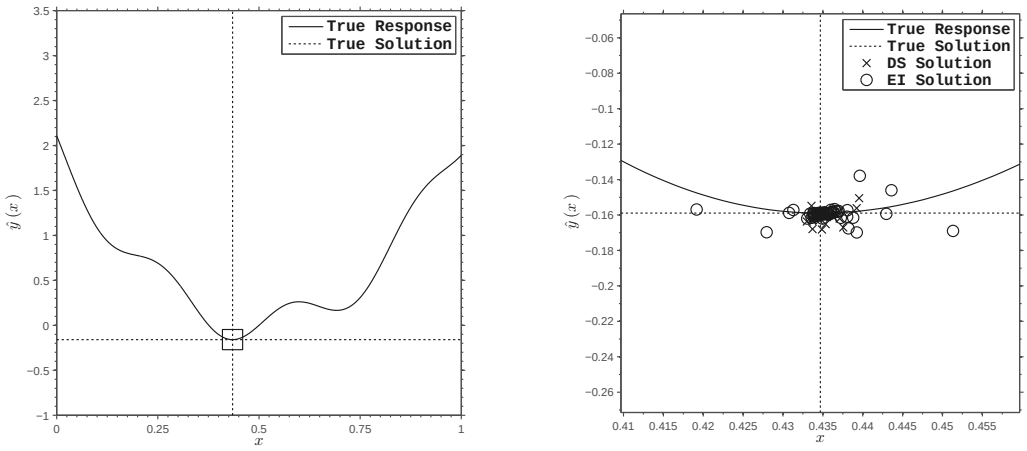
are $Set = [15\ 45\ 135]$.

To ensure the validity of each comparison, the termination criteria used for the expected improvement results is not based on a threshold ratio. Rather, they are based on the same termination criteria as distributed surrogate, the uncertainty of the final distributed solution. This requires a distributed surrogate assessment in each iteration of the expected improvement process and only provides termination criteria without having an impact on infill locations. Since distributed surrogate has the goal of reducing solution uncertainty while expected improvement has the goal of finding a more optimal solution, selecting equivalent termination settings with the use of a threshold ratio is not possible leading to the described alteration.

4.6.1 Global Optimization

Two global optimization example problems are used to demonstrate the distributed surrogate optimization procedure using benchmark problems. The first problem, the 1D problem which has been used to demonstrate parts of the methodology up to the point as well as Expected Improvement, is solved with the distributed surrogate and expected improvement methods using the described batch of initial sample plans. The true response for this problem is shown in 4.21a with the resulting optimum points for both methods shown in 4.21b. Statistics describing the distributed surrogate benchmark are shown in table 4.6 while statistics describing the expected improvement benchmarks are shown in table 4.7. Here, it can be seen that on average, the final amount of truth samples required to terminate the distributed surrogate infilling process, n_{ts}, is nearly 10 samples while the corresponding statistic for expected improvement was nearly 9 samples with negligible differences regarding average final solution, $mean\,(x^*)$ and $mean\,(\hat{y}^*)$. However, the standard deviation regarding the solution point, $std\,(x^*)$, for the distributed surrogate method is nearly one

third that of expected improvement. Another observation worth mentioning is the average number of iterations required, n_k, was nearly 1 iteration greater for expected improvement.



(a) True Response

(b) Benchmark Solution Scatter

Figure 4.21: Benchmark Results for 1D Optimization

Statistic	n_k	n_{ts}	x^*	$\hat{y}^*$
$min\,(\cdot)$	1	8	0.4329	-0.1681
$max\,(\cdot)$	2	13	0.4395	-0.1505
$mean\,(\cdot)$	1.051	9.974	0.4348	-0.1596
$std\,(\cdot)$	0.223	0.986	0.0010	0.0024
$true\,(\cdot)$	—	—	0.4346	-0.1589

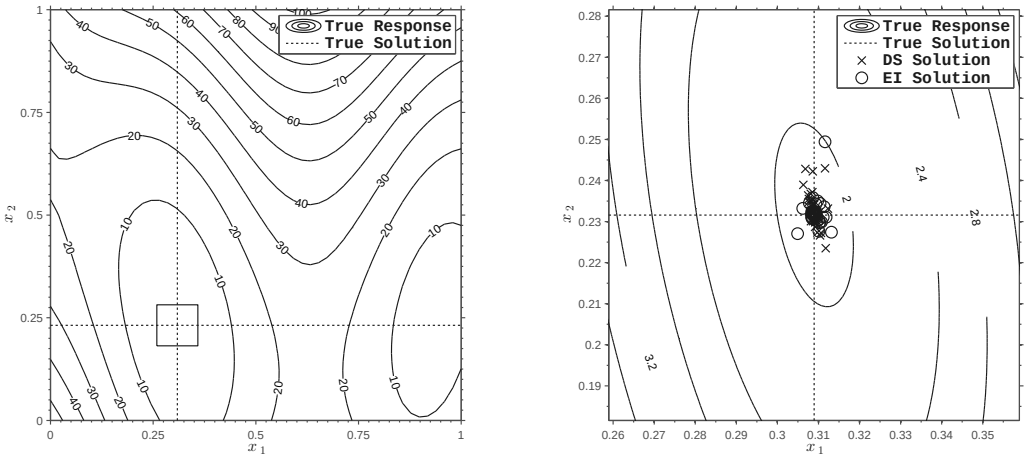
Table 4.6: Distributed Surrogate 1D Optimization Benchmark Results

Statistic	n_k	n_{ts}	x^*	$\hat{y}^*$
$min\,(\cdot)$	0	7	0.4191	-0.1698
$max\,(\cdot)$	4	10	0.4513	-0.1378
$mean\,(\cdot)$	1.740	8.753	0.4352	-0.1591
$std\,(\cdot)$	0.833	0.746	0.0033	0.0037
$true\,(\cdot)$	—	—	0.4346	-0.1589

Table 4.7: Expected Improvement 1D Optimization Benchmark Results

The second problem used to test the global optimization functionality of the distributed surrogate method is a 2D function for which the true response is shown in figure 4.22a.

The distributed solutions found was well as process statistics are shown in figure 4.22b and tables 4.8 and 4.9, respectively. It can be seen the optimal solutions found for both responses agree well with the overall true solution. However, the average number of samples required for the distributed surrogate method was nearly 1.5 samples greater than that of expected improvement. Once again however, the average number of iterations required for distributed surrogate is 1 iteration less.



(a) True Response (b) Benchmark Solution Scatter

Figure 4.22: Benchmark Results for 2D Optimization

Statistic	n_k	n_{ts}	${x_1}^*$	${x_2}^*$	$\hat{y}^*$
$min\,(\cdot)$	1	21	0.3063	0.2235	1.8374
$max\,(\cdot)$	4	28	0.3123	0.2429	2.1204
$mean\,(\cdot)$	1.950	24.55	0.3090	0.2327	1.9464
$std\,(\cdot)$	0.749	1.616	0.0012	0.0041	0.0566
$true\,(\cdot)$	—	—	0.3089	0.2315	1.9557

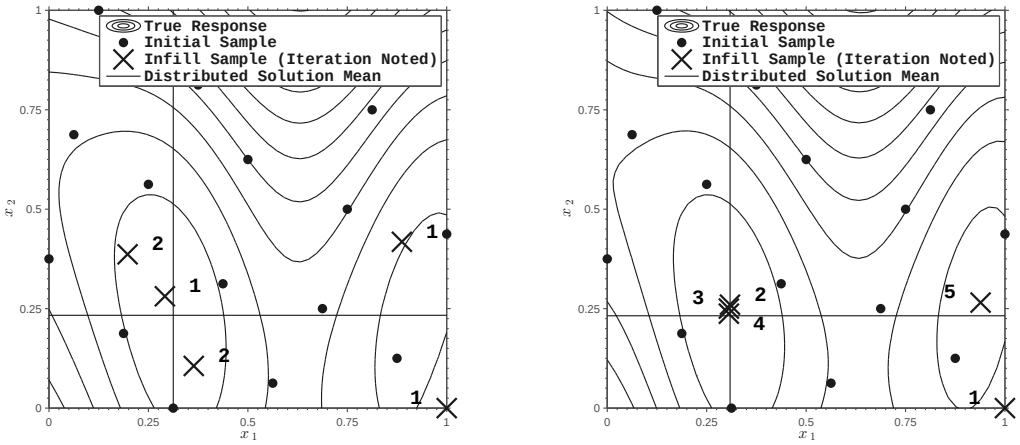
Table 4.8: Distributed Surrogate 2D Optimization Benchmark Results

Statistic	n_k	n_{ts}	${x_1}^*$	${x_2}^*$	$\hat{y}^*$
$min\,(\cdot)$	1	19	0.3049	0.2270	1.8350
$max\,(\cdot)$	5	26	0.3131	0.2494	2.0346
$mean\,(\cdot)$	3.075	23.07	0.3093	0.2321	1.9507
$std\,(\cdot)$	0.971	1.940	0.0014	0.0033	0.0240
$true\,(\cdot)$	—	—	0.3089	0.2315	1.9557

Table 4.9: Expected Improvement 2D Optimization Benchmark Results

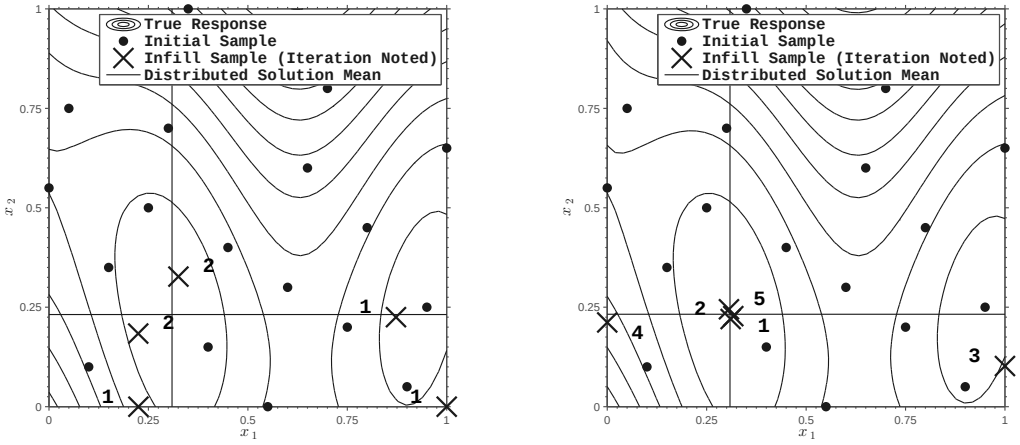
This study provides a comparison between the robustness of both distributed surrogate optimization and expected improvement, however, the efficiency between distributed surrogate and expected improvement is extremely dependent on termination criteria settings. The results shown here reflect those which can be expected by using sensible criteria with respect to the solution spread from a distributed assessment. Although the overall number of truth evaluations required is on average greater for distributed surrogate, it often requires less iterations to arrive at a final solution. The main value of the distributed surrogate method is that it provides information which expected improvement is incapable of approximating such as the solution confidence which is arguably a more meaning termination criteria than a response threshold ratio and the spacing between infills.

To demonstrate the desirably of the selected infill samples by the distributed surrogate method, two initial sample plans used for the 2D problem benchmark results above are presented with the infills selected by both methods. The final infills for the first plan of 17 initial samples are shown in figure 4.23 with the corresponding information for the second plan of 21 initial samples shown in figure 4.24. Upon inspection of each, it can clearly be seen the infills selected by the distributed surrogate method maintain far greater space filling characteristics. In each of the final sample plans for expected improvement, three infills are tightly grouped together near the optimal solution thereby adding redundant information to the surrogate training process. Each final plan shows elements of global exploration, however the infills selected by distributed surrogate maintain good spacing even in the vicinity of the solution.



(a) Distributed Surrogate (b) Expected Improvement

Figure 4.23: Sampling Upon Algorithm Termination, Plan 1



(a) Distributed Surrogate (b) Expected Improvement

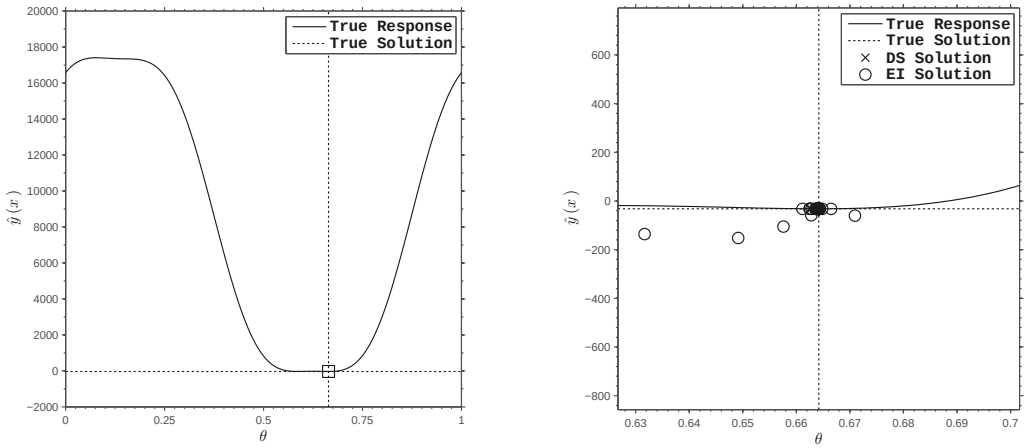
Figure 4.24: Sampling Upon Algorithm Termination, Plan 2

4.6.2 Inverse MPP Search

In addition to the case studies presented for global optimization, three inverse MPP search example problems are also solved using both the distributed surrogate method and expected improvement. For each inverse MPP search, the number of degrees of freedom being optimized is one less than the probabilistic constraint for which it is associated. This is

due to the optimization taking place in n-spherical coordinates with a known radius, the target reliability index, β^t. Since the limit state response is only trying to be minimized, and its final response is not important is RBDO using SORA, the distributed surrogate termination criterion regrading solution response uncertainty is removed, leaving only the spacial point uncertainty criterion.

The true response for the first inverse MPP example is shown in figure 4.25a. The only degree of freedom here, θ, is an angle in polar coordinates which specifies a two dimensional point in standard normal space, u_1 and u_2. Figure 4.25b shows the closeness of all benchmark solutions for this example. Tables 4.10 and 4.11 show the corresponding statistics. It can be seen that the two methods perform in much the same nature with the selected termination criteria, however the final spacial solution standard deviation for distributed surrogate method is shown to be less for this example.



(a) True Response

(b) Benchmark Solution Scatter

Figure 4.25: Benchmark Results for 1D Inverse MPP Search

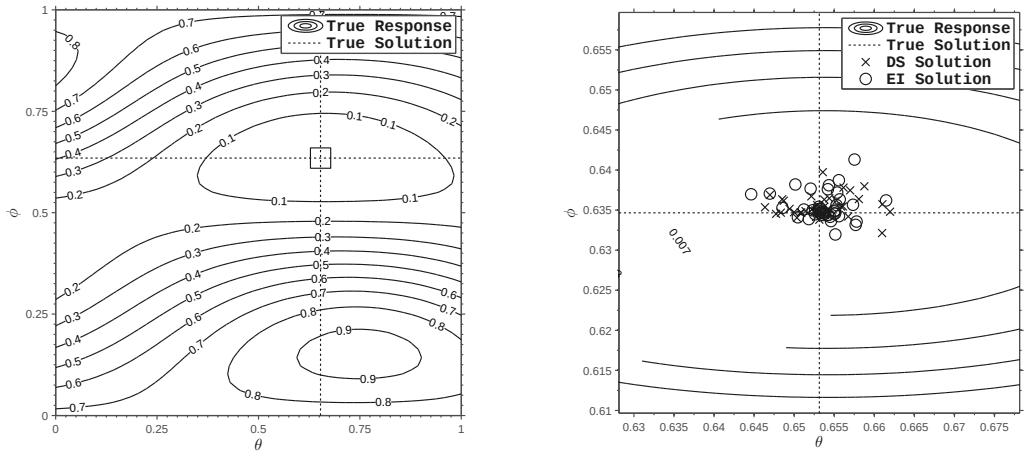
Statistic	n_k	n_{ts}	x^*	$\hat{y}^*$
$min\,(\cdot)$	3	9	0.6626	-32.628
$max\,(\cdot)$	4	13	0.6643	-31.991
$mean\,(\cdot)$	3.753	11.07	0.6641	-32.157
$std\,(\cdot)$	0.433	0.899	0.0002	0.1167
$true\,(\cdot)$	—	—	0.6641	-32.106

Table 4.10: Distributed Surrogate 1D Inverse MPP Search Benchmark Results

Statistic	n_k	n_{ts}	x^*	$\hat{y}^*$
$min\,(\cdot)$	3	7	0.6316	-151.96
$max\,(\cdot)$	6	11	0.6709	-32.099
$mean\,(\cdot)$	4.766	9.883	0.6634	-36.721
$std\,(\cdot)$	0.741	1.012	0.0042	19.992
$true\,(\cdot)$	—	—	0.6641	-32.106

Table 4.11: Expected Improvement 1D Inverse MPP Search Benchmark Results

The second and third inverse MPP search examples are taken from the cantilever RBDO example, each of which have two degrees of freedom, angles θ and ϕ, when transformed to standard normal space. The first is the stress probabilistic constraint described in equation 3.9. The truth response and benchmark solutions are shown in figure 4.26. The statistics for the two methods, shown in tables 4.12 and 4.13 indicate the closeness of the final results.



(a) True Response

(b) Benchmark Solution Scatter

Figure 4.26: Benchmark Results for Cantilever Stress Inverse MPP Search

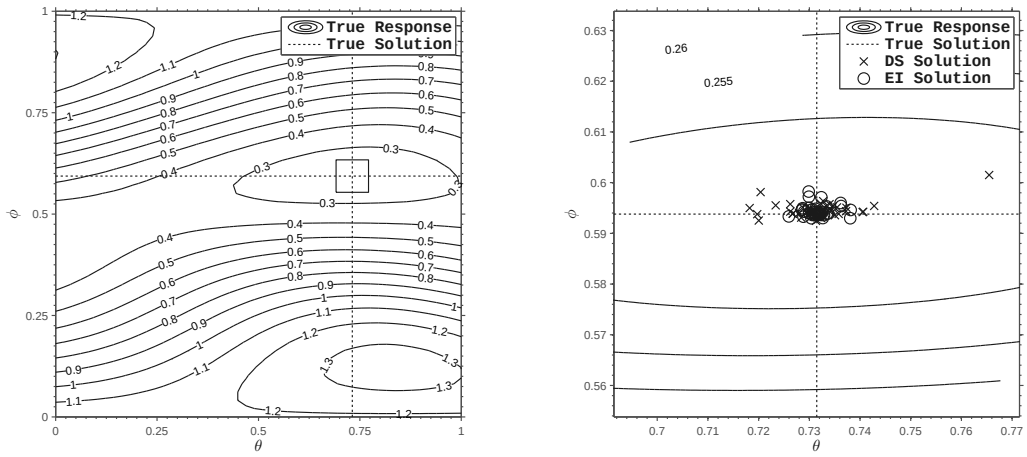
Statistic	n_k	n_{ts}	${\theta_1}^*$	${\phi_2}^*$	$\hat{y}^*$
$min\,(\cdot)$	3	20	0.6463	0.6321	0.0001
$max\,(\cdot)$	7	27	0.6619	0.6397	0.0068
$mean\,(\cdot)$	4.275	24.10	0.6533	0.6354	0.0046
$std\,(\cdot)$	0.960	1.808	0.0037	0.0013	0.0014
$true\,(\cdot)$	—	—	0.6531	0.6346	0.0056

Table 4.12: Distributed Surrogate Cantilever Stress MPP Search Benchmark Results

Statistic	n_k	n_{ts}	${\theta_1}^*$	${\phi_2}^*$	$\hat{y}^*$
$min\,(\cdot)$	3	19	0.6446	0.6319	0.0051
$max\,(\cdot)$	6	26	0.6614	0.6413	0.0058
$mean\,(\cdot)$	4.250	22.25	0.6537	0.6354	0.0055
$std\,(\cdot)$	0.839	1.808	0.0029	0.0017	0.0001
$true\,(\cdot)$	—	—	0.6531	0.6346	0.0056

Table 4.13: Expected Improvement Cantilever Stress MPP Search Benchmark Results

The second inverse MPP search corresponding the cantilever RBDO problem is the deflection probabilistic constraint shown in equation 3.11. The spread in final benchmark solutions for this example is larger along the θ axis than for the stress example for each method used, but is especially evident for the expected improvement results shown in figure 4.27. The statistics for the results for each method's final results are shown in tables 4.14 and 4.15.



(a) True Response

(b) Benchmark Solution Scatter

Figure 4.27: Benchmark Results for Cantilever Deflection Inverse MPP Search

Statistic	n_k	n_{ts}	θ^*	ϕ^*	$\hat{y}^*$
$min\,(\cdot)$	3	19	0.7182	0.5924	0.2373
$max\,(\cdot)$	6	29	0.7654	0.6015	0.2481
$mean\,(\cdot)$	4.225	25.22	0.7318	0.5946	0.2456
$std\,(\cdot)$	0.697	2.154	0.0077	0.0015	0.0017
$true\,(\cdot)$	—	—	0.7314	0.5938	0.2459

Table 4.14: Distributed Surrogate Cantilever Deflection MPP Search Benchmark Results

Statistic	n_k	n_{ts}	θ^*	ϕ^*	$\hat{y}^*$
$min\,(\cdot)$	3	20	0.7259	0.5929	0.2453
$max\,(\cdot)$	9	26	0.7381	0.5982	0.2461
$mean\,(\cdot)$	5.100	23.10	0.7315	0.5943	0.2459
$std\,(\cdot)$	1.277	1.629	0.0024	0.0011	0.0001
$true\,(\cdot)$	—	—	0.7314	0.5938	0.2459

Table 4.15: Expected Improvement Cantilever Deflection MPP Search Benchmark Results

The inverse MPP problem results indicate the robustness of the distributed surrogate method in determining a globally optimal result using only the spacial uncertainty approximation information. Each group of results obtained using different amounts of truth model knowledge located in difference areas of the design space, and the distributed surrogate method

performs competitively with expected improvement with the supplied termination criteria. The trends identified in the global optimization comparison are maintained here, that a greater number of truth samples may be required for the distributed surrogate method, but it generally takes less iterations to arrived at a solution. The characteristic of using less iterations becomes valuable when parallel computing capabilities are at the disposal of the design engineer, facilitating the running of truth responses simultaneously withing the same iteration.

Conclusion and Future Work

Work aiming to provide a more meaningful termination criteria for surrogate based optimization has been conducted. This work follows the development of an analysis tool for surrogate based RBDO, which inspired the need for such criteria. The proposed method, a distributed surrogate method for global optimization, has been developed to address engineering design optimization problems in which the probable spacial solution distribution based on the model prediction uncertainty is needed to quantify solution confidence with response to all degrees of freedom as well as system response. Reasoning behind the Gaussian process perturbation strategy to achieve this has been discussed and outlined though example usage for simple global optimization problems. A study which supports solution uncertainty convergence as more resources are put forth for its approximation is presented and used to justify the resources used in application. Infilling the design space for the improvement of solution certainty is also discussed with an accompanying proposed strategy which uses the distributed solution information previously collected. This infill strategy was developed with short cycle surrogate based optimization in mind, in which multiple supplemental points are evaluated in each of a few algorithm iterations to better utilize available computing resources. Evidence suggesting the robustness of the distributed surrogate algorithm has also been presented with case study results involving the optimization of multiple benchmark problems using different amounts of initial surrogate building information and the regions in which this information is supplied within design space. Since the

distributed surrogate method for optimization operates for the purpose of reducing spacial uncertainty, it is found to be specifically well suited for problems which require a solution point only without the corresponding optimal response. However, the proposed method facilitates the enforcement of solution confidence to a specified degree regarding the response, location, or both if necessary.

Next steps for the continued development of the described algorithm will be work aiming to improve efficiency. During the distributed surrogate assessment, the number of distributed surfaces formed is increased until the standard deviation of the distributed solutions has converged or reached the maximum allowable amount specified. This process is performed even for uncertainty approximations which are not approaching the termination criteria, therefore the continuation of this process until convergence is unnecessary for situations in which the projected converged uncertainty would still exceed the allowable amount. Distributed solutions using less overall distributed surfaces can still provide abundant information for the placement of infill points.

Further work could also be directed at improving the supplemental candidates for problems in higher dimensions. As the number of supplemental candidates increases, the difficulty in locating all the local maximum MSE points increases. This is due to the addition of candidates from previous levels to the building process decreasing the MSE surface values to such a degree that the response becomes unreliable due to numerical noise. An alternative criteria for locating the supplemental points after the first or second level would be useful in maintaining the desired relative spacing filling nature of the distributed infilling process.

Bibliography

[1] Brian M Adams, William Eugene Hart, Michael Scott Eldred, Daniel M Dunlavy, Patricia Diane Hough, Anthony Andrew Giunta, Joshua D Griffin, Monica L Martinez-Canales, Jean-Paul Watson, Tamara Gibson Kolda, et al. *DAKOTA, a Multilevel Parellel Object-oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 4.0 Uers's Manual*. United States. Department of Energy, 2006.

[2] Younes Aoues and Alaa Chateauneuf. Benchmark study of numerical methods for reliability-based design optimization. *Structural and multidisciplinary optimization*, 41(2):277 – 294, 2010.

[3] Daniel W. Apley, Jun Liu, and Wei Chen. Understanding the effects of model uncertainty in robust design with computer experiments. *Journal of Mechanical Design*, 128(4):945 – 958, 2006.

[4] Charles Audet, Andrew J Booker, JE Dennis Jr, Paul D Frank, and Douglas W Moore. A surrogate-model-based method for constrained optimization. *AIAA paper*, 4891, 2000.

[5] Anirban Basudhar and Samy Missoum. Adaptive explicit decision functions for prob-

abilistic design and optimization using support vector machines. *Computers & Structures*, 86(19):1904 – 1917, 2008.

[6] Barron J Bichon, Michael S Eldred, Laura Painton Swiler, Sandaran Mahadevan, and John M McFarland. Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA journal*, 46(10):2459 – 2468, 2008.

[7] Barron J Bichon and John M McFarland. Inverse reliability analysis with egra. In *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 19th AIAA/ASME/AHS Adaptive Structures Conference 13t*, 2011.

[8] Felipe A C. Viana, Victor Pecheny, and Raphael T Haftka. Using cross validation to design conservative surrogates. *Aiaa Journal*, 48(10):2286 – 2298, 2010.

[9] Zhenzhong Chen, Siping Peng, Xiaoke Li, Haobo Qiu, Huadi Xiong, Liang Gao, and Peigen Li. An importance boundary sampling method for reliability-based design optimization using kriging model. *Structural and Multidisciplinary Optimization*, pages 1 – 16, 2014.

[10] Zhenzhong Chen, Haobo Qiu, Liang Gao, Xiaoke Li, and Peigen Li. A local adaptive sampling method for reliability-based design optimization using kriging model. *Structural and Multidisciplinary Optimization*, 49(3):401 – 416, 2014.

[11] K.K. Choi, Byeng D. Youn, and Ren-Jye Yang.

[12] M Daszykowski, B Walczak, and DL Massart. Looking for natural patterns in data: Part 1. density-based approach. *Chemometrics and Intelligent Laboratory Systems*, 56(2):83 – 92, 2001.

[13] Xiaoping Du and Wei Chen. Sequential optimization and reliability assessment method for efficient probabilistic design. *Journal of Mechanical Design*, 126(2):225 – 233, 2004.

[14] Vincent Dubourg, Bruno Sudret, and Jean-Marc Bourinet. Reliability-based design optimization using kriging surrogates and subset simulation. *Structural and Multidisciplinary Optimization*, 44(5):673 – 690, 2011.

[15] Alexander I. J. Forrester and Donald R. Jones. Global optimization of deceptive functions with sparse sampling. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 10 – 12, 2008.

[16] Alexander I. J. Forrester, Andrs Sbester, and A. J. Keane. *Engineering Design via Surrogate Modelling : A Practical Guide*. Progress in astronautics and aeronautics: v. 226. Chichester, West Sussex, England ; Hoboken, NJ : J. Wiley, 2008.

[17] Loic Le Gratiet and Claire Cannamela. Kriging-based sequential design strategies using fast cross-validation techniques with extensions to multi-fidelity computer codes. *arXiv preprint arXiv:1210.6187*, 2012.

[18] S mark Handcock. On cascading latin hypercube designs and additive models for experiments. *Communications in Statistics-Theory and Methods*, 20(2):417 – 439, 1991.

[19] Abraham M Hasofer and Niels C Lind. Exact and invariant second-moment code format. *Journal of the Engineering Mechanics division*, 100(1):111 – 121, 1974.

[20] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T. Hastie, J. Friedman, and R. Tibshirani. *The Elements of Statistical Learning*, volume 2. Springer, 2009.

[21] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455 – 492, 1998.

[22] Chwail Kim and Kyung K. Choi. Reliability-based design optimization using re-

sponse surface method with prediction interval estimation. *Journal of Mechanical Design*, 130(12):121401, 2008.

[23] Régis Lebrun and Anne Dutfoy. A generalization of the nataf transformation to distributions with elliptical copula. *Probabilistic Engineering Mechanics*, 24(2):172 – 178, 2009.

[24] Ikjin Lee, KK Choi, and Liang Zhao. Sampling-based rbdo using the stochastic sensitivity analysis and dynamic kriging method. *Structural and Multidisciplinary Optimization*, 44(3):299 – 317, 2011.

[25] Tae Hee Lee and Jae Jun Jung. A sampling technique enhancing accuracy and efficiency of metamodel-based rbdo: Constraint boundary sampling. *Computers & Structures*, 86(13):1463 – 1476, 2008.

[26] Sankaran Mahadevan and A Haldar. *Probability, reliability and statistical method in engineering design*. John Wiley & Sons, 2000.

[27] Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117 – 129):2, 1978.

[28] Roxanne A Moore, David A Romero, and Christiaan JJ Paredis. Value-based global optimization. *Journal of Mechanical Design*, 136(4):041003, 2014.

[29] Max D. Morris and Toby J. Mitchell. Exploratory designs for computational experiments. *Journal of statistical planning and inference*, 43(3):381 – 402, 1995.

[30] Art B Owen. Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, 2(2):439 – 452, 1992.

[31] James M. Parr, Alexander I. J. Forrester, Andy J. Keane, and Carren M. E. Holden.

Enhancing infill sampling criteria for surrogate based constrained optimization. *Journal of Computational Methods in Science and Engineering*, 12(1):25 – 45, 2012.

[32] Nestor V Queipo, Alexander Verde, Salvador Pintos, and Raphael T Haftka. Assessing the value of another cycle in gaussian process surrogate-based optimization. *Structural and Multidisciplinary Optimization*, 39(5):459 – 475, 2009.

[33] Rüdiger Rackwitz and Bernd Flessler. Structural reliability under combined random load sequences. *Computers & Structures*, 9(5):489 – 494, 1978.

[34] Murray Rosenblatt. Remarks on a multivariate transformation. *The annals of mathematical statistics*, pages 470 – 472, 1952.

[35] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn.

[36] Jian Tu, Kyung K Choi, and Young H Park. A new study on reliability-based design optimization. *Journal of mechanical design*, 121(4):557 – 564, 1999.

[37] Felipe AC Viana, Raphael T Haftka, and Layne T Watson. Why not run the efficient global optimization algorithm with multiple surrogates. In *Proceedings of the 51th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference, AIAA, Orlando, FL, USA. AIAA – 2010 – 3090*, 2010.

[38] Zequn Wang and Pingfeng Wang. A maximum confidence enhancement based sequential sampling scheme for simulation-based design. *Journal of Mechanical Design*, 136(2):021006, 2014.

[39] Byeng D Youn and Kyung K Choi. An investigation of nonlinearity of reliability-based design optimization approaches. *Journal of mechanical design*, 126(3):403 – 411, 2004.

[40] Byeng D. Youn and Kyung K. Choi. A new response surface methodology for reliability-based design optimization. *Computers & structures*, 82(2):241–256, 2004.

[41] Byeng D. Youn, Kyung K. Choi, and Young H. Park. Hybrid analysis method for reliability-based design optimization. *Journal of Mechanical Design*, 125(2):221 – 232, 2003.

[42] Liang Zhao, K. K. Choi, and Ikjin Lee. Metamodeling method using dynamic kriging for design optimization. *AIAA journal*, 49(9):2034 – 2046, 2011.

[43] Liang Zhao, K. K. Choi, Ikjin Lee, and David Gorsich. Conservative surrogate model using weighted kriging variance for sampling-based rbdo. *Journal of Mechanical Design*, 135(9):091003, 2013.

[44] Liang Zhao, Kyung K. Choi, Ikjin Lee, and Liu Du. Response surface method using sequential sampling for reliability-based design optimization. In *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 1171 – 1181. American Society of Mechanical Engineers, 2009.

[45] Xiaotian Zhuang and Rong Pan. A sequential sampling strategy to improve reliability-based design optimization with implicit constraint functions. *Journal of Mechanical Design*, 134(2):021002, 2012.